

THESIS / THÈSE

MASTER EN SCIENCES INFORMATIQUES

Langage de commande pour l'édition des budgets provinciaux

Museme, Kyabutwa

Award date:
1977

Awarding institution:
Universite de Namur

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

Institut d'Informatique

Année académique 1976-1977

Langage de commande
pour l'édition
des budgets provinciaux

KYABUTWA MUSEME

Mémoire présenté en vue de
l'obtention du grade de
Licencié et Maître en Informatique

FNB 16/1977/17

FACULTES
UNIVERSITAIRES
N.-D. DE LA PAIX
NAMUR

Bibliothèque

FMB

16/1977/17

FACULTES UNIVERSITAIRES NOTRE-DAME DE LA PAIX - NAMUR

Institut d'Informatique

Année académique 1976-1977

Langage de commande pour l'édition des budgets provinciaux

KYABUTWA MUSEME

Mémoire présenté en vue de
l'obtention du grade de
Licencié et Maître en Informatique

Page 1 of 1

1977-78

LBS 7856700



University of the Philippines
Division of Education
Provincial Office

1977-78

1977-78

Qu'il nous soit permis, au seuil de notre mémoire, de remercier les personnes suivantes pour leur direction, leurs conseils et leur aide:

Messieurs F.Bodart, qui a proposé le sujet de notre mémoire; Cl.Cherton, Directeur de ce même mémoire, pour ses conseils et ses remarques; Th.Sow, pour ses suggestions et W.Paulus, pour son aide à la rédaction de notre travail.

Que les Autorités de la Province de Namur, les membres de l'ASBL CIGER, ceux de l'Institut d'Informatique et des Facultés Universitaires Notre-Dame de la Paix à Namur trouvent également ici l'expression de nos plus vifs remerciements pour leur assistance et leur collaboration à notre formation d'informaticien.

TABLE DES MATIERES

Page

	Page
2.2. Structure d'accès -----	17
2.2.1. Rôle de la structure d'accès -----	17
2.2.2. Relations d'accès entre les différents objets -----	17
CHAPITRE III: ETAT ACTUEL DE L'INFORMATISATION -----	20
3.1. Aide à l'Elaboration du budget -----	20
3.1.1. Création du budget provincial -----	20
3.1.2. Création du budget analytique -----	21
3.2. Les mises à jour des budgets -----	21
3.2.1. Mises à jour du budget provincial -----	22
3.2.2. Mises à jour du budget analytique -----	23
3.3. Editions des budgets -----	23
3.3.1. Le budget provincial -----	
3.3.1.1. Etats imprimés -----	24
3.3.1.2. Caractéristiques de l'édition -----	25
3.3.2. Le budget analytique -----	26
3.3.2.1. Etat imprimé -----	26
3.3.2.2. Caractéristiques de l'édition -----	26
3.4. Problème à résoudre -----	33
3.5. Conclusion -----	34
3.6. Solution proposée -----	34
CHAPITRE IV: DESCRIPTION DU LANGUAGE -----	36
4.1. Généralités -----	36
4.1.1. Introduction -----	36
4.1.2. Comparaison du langage avec les langages de consultation de données ----	37
4.1.2.1. Le langage SOCRATE -----	
4.1.2.2. L'IQF (Interactive Query Facility) -----	39
4.1.3. Objectifs du langage -----	41
4.2. Eléments nécessaires à une édition -----	42
4.2.1. Questions susceptibles d'être posées par l'utilisateur -----	

4.2.1.1. Questions relatives au budget	provincial -----	42
4.2.1.2. Questions relatives au budget	analytique -----	43
4.2.2. Analyse de la forme des questions	-----	43
4.2.2.1. Description de quelques questions	-----	43
4.2.2.2. Méthode de traitement des questions	---	49
4.2.3. Conclusion de la description des	questions ---	50
4.2.3.1. Opérateurs à mettre à la disposition	de l'utilisateur -----	50
4.2.3.2. Paramètres à faire figurer dans une	commande -----	
CHAPITRE V: TYPOLOGIE DES EDITIONS	-----	53
5.1. Critères de classification des états	imprimés -----	53
5.1.1. Volume des informations à éditer	-----	53
5.1.2. Temps d'attente souhaité par l'utili-	sateur ---	53
5.1.3. Degré de complexité de la mise en pages	-	54
5.1.4. Caractère cyclique de l'édition	demandée -----	55
5.2. Formats d'édition	-----	56
5.2.1. Les formats standards	-----	
5.2.2. Les formats libres	-----	57
5.2.2.1. Le mode 1	-----	58
5.2.2.2. Le mode 2	-----	58
5.2.2.3. Le mode 3	-----	
5.2.2.4. Etats imprimés en formats libres:	quelques exemples -----	60

	Page
CHAPITRE VI: PROPOSITION DE SPECIFICATIONS DU LANGAGE -----	66
6.1. Notations formelles -----	66
6.2. Les commandes du langage -----	66
6.2.1. La commande DEMANDE -----	67
6.2.2. La commande TOTAL -----	68
6.2.3. La commande MODIF -----	70
6.2.4. La commande TRANSF -----	71
6.2.5. La commande AFFICHE -----	72
6.3. Justifications et analyse des	
commandes choisies -----	73
6.3.1. Commandes de sélection -----	73
6.3.2. Commandes de traitement -----	77
6.3.3. Pourquoi les formats libres? -----	82
6.3.4. Messages du système -----	86
6.3.5. Grammaire formelle du langage -----	87
6.3.6. Exemples de formation des commandes --	90
6.4. Considérations relatives à l'implémen-	
tation -----	93
6.4.1. Solutions possibles -----	93
6.4.2. Méthode de conception -----	95
6.4.3. Réalisation des accès aux données -----	96
6.4.4. Exécution des opérations demandées ----	97
6.4.5. Définition des états imprimés -----	99

CONCLUSION

TABLE DES MATIERES

INTRODUCTION

Les services informatiques mettent souvent à la disposition des gestionnaires une masse importante d'informations sous forme d'états imprimés.

Or les utilisateurs ne désirent pas nécessairement disposer en une seule fois de toutes les informations auxquelles ils peuvent accéder. Ils souhaitent au contraire choisir parmi un ensemble de données, celles qu'ils considèrent comme utiles à leurs préoccupations du moment.

Le but de ce travail est de réaliser un langage qui permet aux gestionnaires d'obtenir à la demande, les renseignements dont ils ont exactement besoin.

Ce langage ne devrait pas prétendre à une trop grande généralité car nous nous sommes limités à un champ d'application bien défini, à savoir les budgets de la Province de Namur.

Puisque, comme nous le verrons dans la suite, on ne peut pas déterminer à l'avance quelles sont les informations dont aura besoin le gestionnaire, la première étape de notre travail a été d'étudier les applications de gestion relatives aux budgets.

Ceci nous a permis d'analyser quelques questions dont les gestionnaires auraient besoin de connaître la réponse.

Nous avons ensuite défini les caractéristiques que doit présenter le langage en tenant compte des besoins qui sont apparus dans les applications examinées.

Nous avons retenu les caractéristiques suivantes:

- un langage simple, c'est à dire pouvant être utilisé directement par un utilisateur non-informaticien;
- un langage conçu pour des applications d'aide à la décision et offrant des possibilités d'utilisation conversationnelle; ceci permettrait aux gestionnaires d'obtenir des réponses dans les délais les plus brefs;
- un langage présentant des possibilités d'évolution, c'est à dire pouvant être adapté constamment au développement des applications.

Enfin, connaissant les applications et les caractéristiques du langage, nous avons proposé des commandes et procédé à la définition d'une syntaxe.

Dans l'état actuel du travail, nous n'avons pu passer au stade de la réalisation pratique sur ordinateur; nous resterons donc au niveau de "l'analyse du problème".

CHAPITRE I : DESCRIPTION DES DIFFERENTS BUDGETS

=====

1.1. Le budget provincial (1)

1.1.1. Nature du budget provincial

Le budget Provincial est régi par l'arrêté royal portant règlement général sur le contrôle des engagements de dépenses et sur la comptabilité des provinces.

C'est donc un budget légal, contrairement au budget analytique qui découle d'une organisation interne de gestion, propre à la Province de Namur.

Le fait que le budget provincial soit légal impose certaines contraintes d'organisation et de structure dont doit tenir compte l'informaticien.

1.1.2. Contenu du budget provincial

Le budget provincial comprend trois volets:

- le budget ordinaire: ce sont des prévisions en dépenses et en recettes de sommes nécessaires pour le fonctionnement de la Province.
- le budget extraordinaire: c'est le budget prévu pour les investissements de la Province.
- le budget pour ordre: c'est l'ensemble des prévisions d'opérations sur des fonds qui ont reçu une affectation particulière. Il comprend notamment les opérations effectuées pour compte de tiers ou de services étrangers à la Province. (ex: le fonds des pensions).

Dans l'ensemble, on peut dire que le droit budgétaire applicable en matière de budget de l'Etat, l'est aussi en ce qui concerne le budget provincial. Un parallélisme assez complet

existe dans l'élaboration, le vote, les corrections en cours d'exercice, l'exécution et le contrôle des budgets.

1.1.3. Elaboration du budget provincial

Le projet de budget provincial est établi par la Députation Permanente du Conseil Provincial. Il est dressé d'après les résultats du compte de l'année pénultième, les prévisions du budget de l'année en cours, les propositions dûment justifiées des établissements et services provinciaux ainsi que tous les éléments établissant les droits et les obligations de la Province.

Ce projet est appuyé en outre de documents permettant d'établir le bien-fondé des crédits de recettes ou de dépenses et d'éclairer le Conseil Provincial et l'autorité de tutelle sur la situation financière de la Province.

On voit déjà apparaître ici l'importance de documents qui doivent être édités. Comme nous le verrons plus loin, cette importance existe à tous les autres niveaux de décision. Cette étape est assez importante parce que la valeur du budget, et son équilibre notamment, dépendent en partie de l'exactitude des prévisions faites.

1.1.4. Vote et approbation du budget provincial

Le Conseil Provincial, saisi de ce projet de budget, arrête les comptes de l'exercice écoulé et vote le budget.

La loi provinciale exige que les délibérations et les résolutions du Conseil Provincial arrêtant le budget soient transmises sans délai au Ministère de l'Intérieur pour être soumises à l'approbation du Roi avant d'être mises à exécution.

1.1.5. Correction des autorisations budgétaires

Pendant l'application du budget, la loi permet certaines corrections. Celles-ci sont d'autant plus nécessaires qu'en dehors même de toute erreur, des événements nouveaux peuvent modifier les prévisions budgétaires. Ces erreurs ou insuffisances doivent être surtout corrigées à l'égard des autorisations de dépenses, car elles pourraient empêcher la Province de fonctionner et de faire face à des dépenses nécessaires et urgentes.

On distingue deux catégories de corrections:

1.1.5.1. les modifications budgétaires

Ce sont des mises à jour à entreprendre sur les articles (a) du budget provincial. Il s'agit des réductions ou des augmentations des crédits budgétaires déjà accordés aux services provinciaux.

Ces modifications (9 au maximum pour un article budgétaire) portent soit sur le nombre d'articles du budget, soit sur le contenu des articles.

1.1.5.2. les transferts budgétaires

On appelle transfert l'utilisation d'un crédit budgétaire à une fin autre que celle à laquelle il avait été affecté lors du vote du budget (2).

Il existe à l'intérieur du budget provincial deux articles intitulés "crédit destiné à pallier l'insuffisance de crédits budgétaires pour des dépenses obligatoires". Ces articles spéciaux servent d'origine à ces mouvements de transfert. L'un sert de réserve aux articles du budget ordinaire, l'autre aux articles du budget extraordinaire.

(a) article est utilisé ici dans le sens de rubrique comptable

Tant en ce qui concerne les modifications que les transferts, les résolutions du Conseil Provincial modifiant le budget sont établies d'après un modèle faisant connaître, pour chacun des budgets ordinaire et extraordinaire, outre les changements à apporter, la situation des prévisions du budget avant et après modification.

Ces résolutions sont transmises au Ministère de l'Intérieur pour approbation (1) .

1.1.6. L'exécution du budget

Le budget provincial, une fois voté et approuvé , entre dans la phase d'exécution. Celle-ci est assurée par un ensemble de procédures administratives qui permettent d'effectuer les dépenses et les recettes autorisées par le budget!

Nous n'avons pas à analyser ces procédures dans leurs détails techniques, mais en résumé, on peut dire que l'exécution du budget ne s'effectue pas de la même manière pour les recettes que pour les dépenses.

1.1.6.1. exécution des dépenses

On distingue quatre stades dans l'opération de dépenses:
l'engagement, l'ordonnancement, la liquidation et le paiement.

Cette distinction repose sur plusieurs fondements: - assurer la clarté et la précision des opérations de dépenses, en permettant d'en séparer nettement les diverses phases;

- permettre de déterminer le moment exact à partir duquel la Province est engagée;

- permettre d'établir une sorte de séparation des pouvoirs à l'intérieur des opérations de dépenses, en confiant leurs diverses phases à des autorités différentes;

- l'engagement est le fait générateur de la dépense provinciale;
- la liquidation est la reconnaissance de la réalité et de la légalité d'une dépense, de même que son imputation à charge du crédit budgétaire adéquat;
- l'ordonnancement des dépenses constitue l'ordre de payer adressé par l'ordonnateur au comptable;
- le paiement de la dépense par le comptable .

1.1.6.2. l'exécution des recettes

Il faut un titre légalement établi (facture ou déclaration de créance) pour opérer une recette.

Ces titres sont établis à l'occasion de prestations, fournitures ou travaux effectués par les établissements et services provinciaux pour compte de tiers ou d'autres services provinciaux.

Les recettes sont effectuées par l'intermédiaire d'agents comptables de l'Etat ou de receveurs spéciaux.

1.2. Le budget analytique (5)

1.2.1 Notion de budget analytique

Le budget analytique de la Province de Namur constitue une prévision de dépenses (ou recettes) sur un an:

- pour une nature de dépenses (ou recettes) données, celle-ci appartient à un service donné (5);
- ce budget n'est pas réglementé par une loi;
- il a été spécialement aménagé pour faciliter l'examen des conditions internes de fonctionnement de la Province et pour donner aux responsables provinciaux

toutes les informations utiles pour les aider à prendre des décisions.

Il devrait (5) permettre notamment: - l'analyse des charges et une observation de leur évolution;

- la comparaison des charges aux prévisions;

- l'analyse des écarts entre les charges réelles et les prévisions.

1.2.2. Intérêt du budget analytique

Le budget analytique de la Province répond à un certain souci de décentralisation de gestion.

De manière tout à fait générale, une gestion budgétaire se fait en trois étapes chaque fois que les gestionnaires se trouvent confrontés à une décision (4), à savoir:

- la prévision: étude préalable de la décision, possibilité et volonté de l'accomplir;
- la budgétisation: définition des objectifs et des moyens propres à les atteindre;
- le contrôle: écarts obtenus entre l'objectif et la réalisation, explication et exploitation de ces écarts.

Dans une gestion très centralisée, l'accomplissement optimal de ces trois étapes devient difficile; il existe trente deux services provinciaux dispersés géographiquement; la nature et les dimensions de différents services sont très diversifiées.

D'où l'intérêt de confier à divers chefs de services certaines responsabilités de gestion.

A tout moment, les documents budgétaires rapprochant prévisions et réalisations peuvent être établis, il devient alors aisé au niveau intéressé de juger l'action des subordonnés .

1.2.3. Le contenu du budget analytique

Le budget analytique est une ventilation du budget provincial ordinaire (a).

En d'autres termes, l'alimentation du budget analytique est obtenue par les crédits inscrits au budget provincial. Aucun article analytique ne pourra être créé sans que le crédit disponible de l'article du budget provincial dont il est issu soit suffisamment approvisionné. Ce crédit disponible est appelé "solde non ventilé en comptabilité analytique".

Nous appellerons "article père", l'article du budget provincial dont un ou plusieurs articles analytiques sont issus.

1.2.4. Transferts affectant le budget analytique

Il arrive qu'en cours de gestion, des mouvements de transferts influencent le contenu du budget analytique. Ces transferts peuvent être de trois natures différentes et répondent aux besoins d'effectuer des "corrections" sur le budget analytique.

Le premier type de transfert s'effectue du crédit disponible de l'article du budget provincial vers un article du budget analytique.

Un transfert peut avoir lieu dans le sens inverse, c'est à dire un transfert du crédit d'un article du budget analytique vers son article père du budget provincial. Ce mouvement a pour effet d'augmenter le crédit disponible à l'article du budget provincial.

Enfin, un dernier type de transfert peut se réaliser entre deux articles du budget analytique possédant le même article

(a) Les budgets extraordinaire et pour ordre sont gérés par l'Administration Centrale de la Province; ils ne sont pas ventilés en budget analytique, celui-ci est un budget décentralisé.

père dans le budget provincial. Ce mouvement permet certains aménagements de gestion à l'intérieur d'un même service provincial.

1.3. Conclusion

Le budget provincial relève du domaine des finances publiques, de la loi sur le contrôle général des engagements et de la comptabilité des provinces, matières vastes qui dépassent le cadre de cette étude.

Dans ce chapitre, nous nous sommes efforcés de résumer les dispositions budgétaires qui nous paraissaient être les plus importantes pour notre travail.

Nous espérons que cette démarche facilitera la compréhension de la suite du travail, notamment la nature et la justification de plusieurs applications informatiques qui seront abordées.

CHAPITRE II: "SOUS-SCHEMA BUDGETS": Différents niveaux de conception

=====

Nous ne pouvons pas proposer un langage d'édition sans dire un mot des fichiers, car c'est autour d'eux que doivent s'organiser les différents traitements.

Dans un système intégré de gestion, on peut regrouper les informations sous forme de bases de données. Les objectifs assignés à une telle démarche sont de deux types:

- modélisation d'un certain "monde réel" sur lequel fonctionnera une organisation;
- optimisation de la fonction disponibilité des données dans un système informatique (6).

Un des rôles assignés à une base de données est de résoudre un problème de communication entre les utilisateurs (gestionnaires) de résultats et la masse d'informations enregistrées. Ce rôle exige l'existence d'un interface de description des données en termes indépendants de toute solution informatique, qui puisse être compris par le gestionnaire.

C'est dans ce contexte que nous décrivons ci-après un sous-schéma (a) contenant toutes les informations relatives aux budgets analytique et provincial. Ce sous-schéma exprime un certain nombre de besoins en informations qui seront utilisées lors de l'analyse des applications.

(a) nomenclature CODASYL pour désigner un sous-ensemble de données dont un ensemble d'applications a besoin.

2.1. Structure conceptuelle (6)

Dans tous les processus d'analyse devant conduire à une réalisation, la première phase (structure conceptuelle) a pour rôle la délimitation du domaine d'application de la réalisation et la réduction du cahier des charges pour la phase suivante.

Cette structure est le point d'accord entre informaticiens et utilisateurs. Elle doit contenir:

- la définition des informations du système et des relations qui les lient;
- les propriétés qualitatives de ces informations;
- les propriétés quantitatives de ces informations;
- l'expression des besoins connus sur les informations décrites;
- les propriétés quantitatives de ces besoins.

Grâce aux contacts multiples que nous avons eus avec le personnel du CIGER et en consultant le dictionnaire des informations (5), nous avons élaboré une structure conceptuelle des données relatives aux budgets.

2.1.1. Définitions des informations

Compte tenu des applications connues, nous avons distingué les types d'objets: article budgétaire, article analytique, transferts, modifications, fonctions, nature économique.

.....
article budgétaire: l'ensemble des réalisations de l'objet article budgétaire constitue le fichier "budget provincial".

A cet objet sont associées les caractéristiques suivantes:

- exercice: année civile à laquelle se rapporte le budget établi;
- nature du budget : code permettant de savoir si l'article du budget appartient au budget ordinaire, extraordinaire ou pour ordre;

- code fonction: code permettant d'identifier un article du budget fonctionnel;
- code économique: détermine la nature de la dépense ou de la recette;
- numéro d'ordre: un numéro permettant de distinguer les articles ayant un même code économique et fonctionnel;
- nature d'un article: chaque article budgétaire peut avoir deux natures: être une dépense ou constituer une recette;
- caractère de crédit: un code permettant de savoir si le crédit consenti à un article est obligatoire ou facultatif;
- solde non ventilé en comptabilité analytique: la différence entre le crédit d'un article du budget provincial et la somme des crédits versés sur chacun des articles du budget analytique lui correspondant;
- chapitre: une subdivision du budget dans la classification officielle;
- crédit budgétaire initial: montant attribué par le Conseil Provincial en début d'exercice pour faire face à un type de dépenses (ou de recettes) défini par l'article du budget;
- crédit mis à jour: crédit prévu à un article du budget et mis à jour après une modification budgétaire ou une décision du Ministère de l'Intérieur.

.....
article analytique: l'ensemble des réalisations de l'objet
article analytique constitue le fichier "budget analytique".

A cet objet sont associées les caractéristiques suivantes:

- exercice: année civile à laquelle se rapporte l'exécution du budget;
- code service: code attribué à un service dans les plans comptables de la Province;
- code section: code attribué à une section d'un service dans les plans comptables;
- code nature de charges: code attribué à une nature de charges reprise dans les plans comptables analytiques des établissements provinciaux;
- crédit budgétaire analytique initial: partie d'un crédit budgétaire global affecté à une section et/ou une nature de charges d'un service, à l'ouverture du budget;
- crédit budgétaire analytique mis à jour: partie du crédit budgétaire global mis à jour affecté à une section et/ou à une nature de charges et pouvant subir des modifications internes dans un même article budgétaire global (5);
- transferts: transferts (montants) d'une partie de crédit d'un article budgétaire analytique au solde non ventilé en comptabilité analytique de l'article du budget provincial dont l'article analytique est issu;
- nom du service provincial: nom d'un service tel qu'il est repris dans les plans comptables;
- nom de la section: nom de la section défini dans les plans comptables provinciaux;
- nature de l'article: code déterminant si l'article est une dépense ou une recette;

- nature de charges: dénomination des natures de charges reprises dans les plans comptables des établissements provinciaux.

Objet "modifications (budgétaires)": A un objet modifications
seront associées les caractéristiques suivantes:

- numéro: le numéro d'ordre de la modification budgétaire, se rapportant au rang d'une réunion de la Députation Permanente ou du Conseil Provincial;
- date: la date à laquelle est intervenue la modification;
- montant: la modification en plus ou en moins apportée au crédit budgétaire d'un article.

Objet "transferts (budgétaires)": A un transfert sont associées les caractéristiques suivantes (a):

- numéro: numéro d'ordre du transfert budgétaire;
- montant: la valeur en francs du crédit transféré;
- date: date à laquelle a eu lieu le transfert.

Objets "fonction" et "nature du crédit":

.....
"Pour que les budgets et comptes puissent être rendus utilisables comme instruments d'informations, il faut qu'ils soient adaptés aux exigences du traitement direct par l'ordinateur. C'est pourquoi ils sont pourvus de codes statistiques." (11)

Objet FONCTION: une fonction est une subdivision du budget économique et fonctionnel (b). A un objet "fonction" sont associés: - un code statistique
- un libellé: le nom de la fonction.

(a) Transfert d'un crédit d'un article de réserve vers un autre article du budget provincial.

(b) voir en annexe 1 la table des fonctions et des natures économiques.

Objet "nature du crédit" (a): C'est la nature des crédits telle qu'elle est définie dans le budget économique et fonctionnel . A une nature du crédit sont associées:

- un code statistique
- un libellé: nom du groupe économique auquel l'article appartient

2.1.2. Relations sémantiques entre les différents objets (b)

Les relations sémantiques que nous allons définir reposent entièrement sur les applications décrites aux paragraphes 3.1 à 3.3. Lorsque nous associerons un type d'objets à un autre, nous associerons implicitement ce dernier au premier.

- La relation (1): article budgétaire vers article analytique. C'est une relation nécessaire pour notre problème, car elle répond à la nécessité de résoudre des problèmes du genre: pouvoir déterminer comment un article du budget a été ventilé en budget analytique.

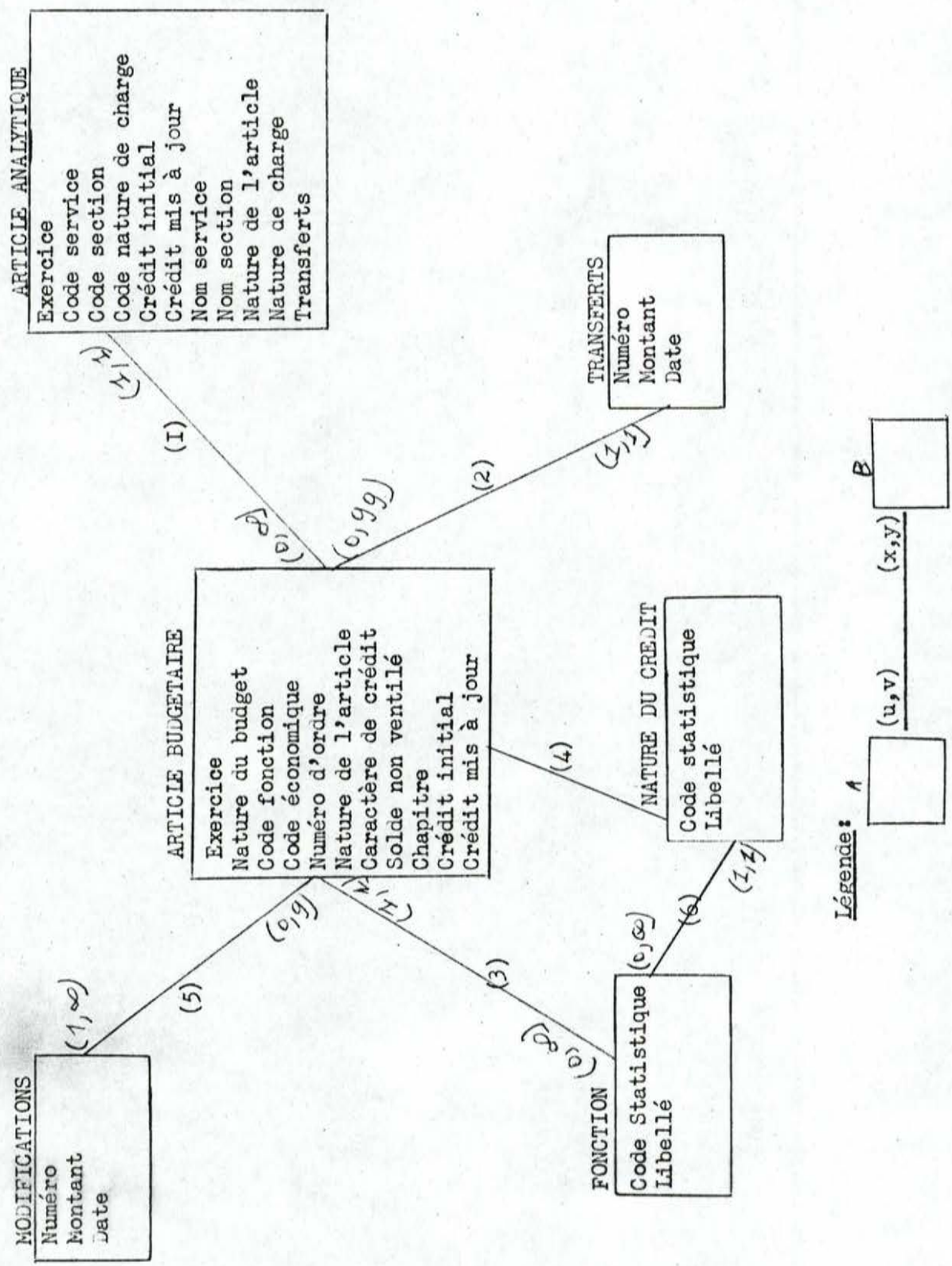
-La relation (2): article budgétaire vers transferts. Exemple d'utilisation: on s'intéresse à tous les transferts ayant influencé un article du budget déterminé.

-la relation (3): article budgétaire vers fonction. C'est une relation importante pour notre problème car elle permet d'effectuer des totalisations des crédits par fonction.

(a) voir table des codes économiques en annexe 2

(b) voir diagramme en page 15.

Grapho de structure



- La relation (4): article budgétaire vers nature du crédit. C'est également une relation intéressante pour nous, car les utilisateurs demandent souvent à connaître les totaux des crédits budgétaires par nature du crédit.
- La relation (5): article budgétaire vers modifications. C'est une relation qui concerne tous les mouvements de modification légale pouvant influencer un article budgétaire.

2.1.3. Propriétés qualitatives des informations décrites

(voir diagramme p. 15)

- A un article budgétaire correspond 0 ou plusieurs articles analytiques.
- A un article budgétaire ne peut correspondre que 0 modification budgétaire au minimum et 9 au maximum.
- Le nombre de transferts par article budgétaire ne peut dépasser 99.
- A un article analytique correspond 1 article père au minimum et 1 au maximum.
- A une fonction et à une nature du crédit peuvent correspondre au minimum 0 article budgétaire et au maximum un nombre indéterminé d'articles budgétaires.
- A un article budgétaire ne peut correspondre au minimum qu'une fonction et qu'une nature du crédit et au maximum qu'une fonction ou qu'une nature du crédit.
- A un transfert ou à une modification ne peut correspondre au minimum qu'un article budgétaire et au maximum qu'un article budgétaire.

Ces propriétés qualitatives sont décrites car lors des mises à jour, le système doit pouvoir vérifier leur validité.

2.2. La structure d'accès

2.2.1. Rôle de la structure d'accès

Le rôle de la structure d'accès est de décrire des chemins d'accès aux données. Tous les résultats attendus doivent pouvoir être extraits de cette structure.

Compte tenu du fait que la plupart des informations sont utilisées par plusieurs applications, la structure d'accès joue un rôle d'optimisation globale de l'ensemble des programmes (6).

Les accès définis dans cette structure sont les seuls accès auxquels pourra faire appel l'utilisateur.

2.2.2. Les relations d'accès entre les différents types d'objets (voir le graphique de la structure d'accès p.19)

Les arcs orientés représentent un chemin d'accès entre deux types d'objets.

Les objets de type article budgétaire et de type article analytique pourront être atteints au moyen d'une clé d'accès comme indiqué sur le diagramme.

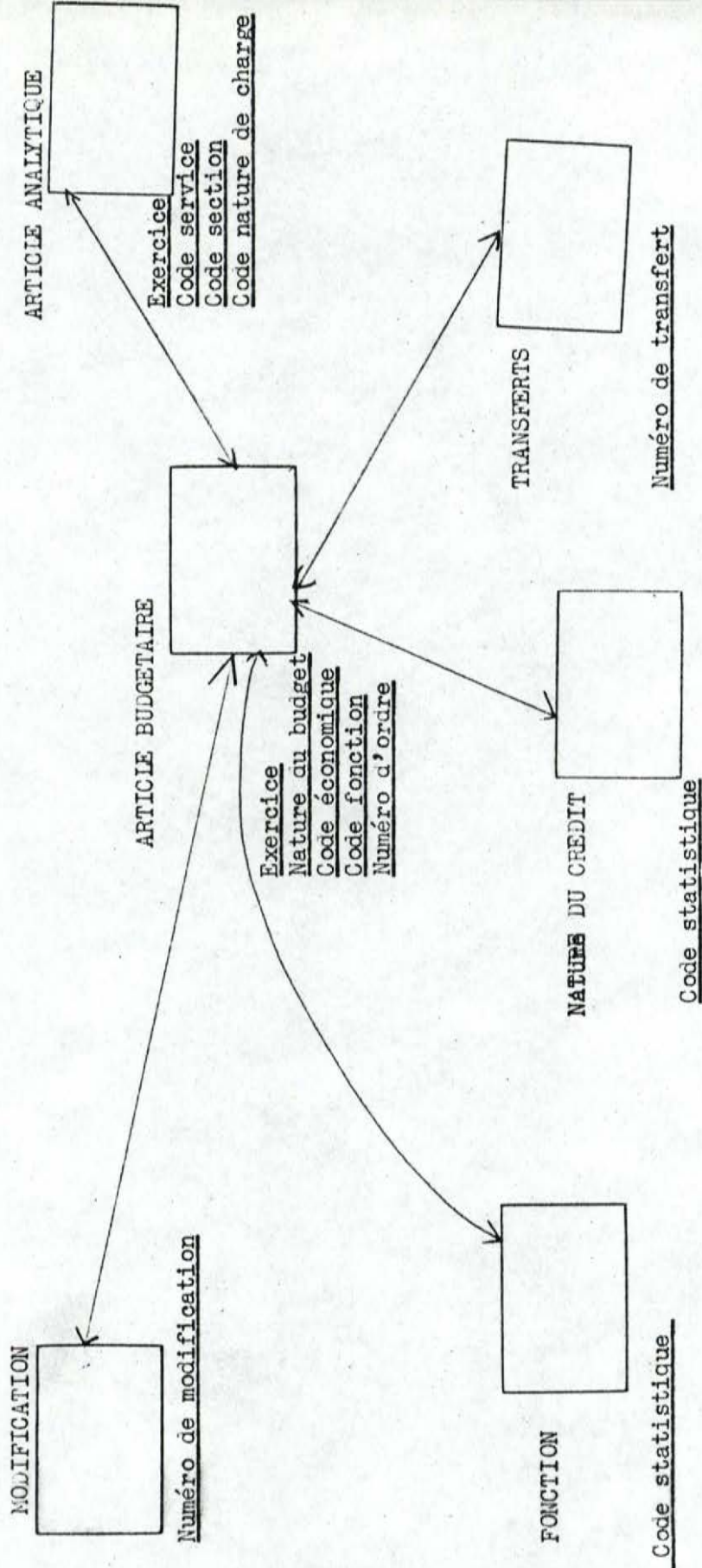
Pour notre travail, les relations d'accès décrites sur ce diagramme seront les seuls chemins suivis par l'utilisateur pour tout travail sur les données.

Il y a toujours plusieurs structures possibles et dans le choix de construction de la structure d'accès, il faut faire un compromis entre les performances attendues et les prises en compte automatiques de certaines contraintes.

Le but de ce travail n'étant pas la construction d'une structure d'accès, nous nous sommes efforcés de proposer une

structure d'accès simple, qui tienne compte des problèmes à résoudre et de la structure existant actuellement (a). C'est un choix que nous avons fait et nous croyons qu'il répond aux problèmes à résoudre.

(a) Dans le système informatique actuel, l'ensemble des occurrences des objets de type article budgétaire et article analytique sont des fichiers séquentiels indexés.



Légende:

→ Chemin d'accès

Exercice == Clé d'accès

: Type d'objet

CHAPITRE III: ETAT ACTUEL DE L'INFORMATISATION

=====

Dans son état actuel, la gestion automatisée des budgets provinciaux se présente non pas comme une application limitée, mais comme un domaine interférant avec plusieurs applications donnant lieu à de nombreux traitements informatiques. Pour ne citer qu'un exemple, la comptabilité des engagements est informatisée et cette application a beaucoup d'influence sur la gestion des budgets.

Une description exhaustive et détaillée de ces applications ne s'impose pas ici. Nous limitons notre examen aux applications relevant uniquement de la gestion des budgets et utilisant le sous-schéma décrit à la page 15. Ce sont les seules applications qui intéressent notre travail.

3.1. Aide à l'élaboration du budget

3.1.1. Création du budget provincial

Le système informatique est actuellement utilisé comme aide à la décision lors de l'élaboration du budget.

Le principe peut se résumer comme suit: à partir du budget provincial (fichier) mis à jour de l'année en cours, on produit une projection pour l'année suivante. Sur base de ce budget, les responsables pourront prendre la décision de modifier, de supprimer ou d'ajouter des articles.

Après chaque réunion, les modifications apportées à l'image précédente seront introduites dans le système et celui-ci éditera une nouvelle image du budget pour l'année suivante.

Cet apport de l'informatique permet aux responsables provinciaux de mesurer l'incidence de diverses options de dépenses ou de recettes sur le budget provincial.

3.1.2. Création du budget analytique

Pour qu'un article du budget analytique soit créé, il faut qu'il existe un crédit disponible dans le "s~~old~~e nonventilé en comptabilité analytique d'exploitation" de l'article père au budget provincial.

Le procédé actuel de création du budget analytique est le suivant: - après la réouverture (a) officielle du budget provincial, le système réouvre automatiquement le budget analytique avec les valeurs de crédit du budget analytique existantes;

- lorsque les crédits disponibles du budget provincial ne sont pas suffisamment approvisionnés pour couvrir les crédits analytiques correspondants, ces derniers sont automatiquement refusés par le système;

- après les aménagements opérés par les responsables et les mises à jour nécessaires effectuées, plusieurs états imprimés sont édités et comparés jusqu'à l'obtention d'un budget analytique définitif.

3.2. Les mises à jour des budgets

Actuellement, les mises à jous sont les applications les plus fréquentes au niveau de l'exploitation des fichiers budgets. En effet, les gestionnaires sont ainsi certains que pour toute application de gestion sur ces fichiers, les traitements sont effectués sur des données correspondant à l'état le plus récent des informations.

Examinons à présent les mises à jour réalisées par le système actuel par type de budget.

(a) reprise pour l'année en cours du budget mis à jour de l'année précédente.

3.2.1. Mises à jour du budget provincial (5)

- Création d'un article du budget provincial: c'est une opération classique de création. On crée l'article dans le fichier budget et le système devra toujours garder trace du montant (crédit) de l'article lors de sa création.
- Création d'un article par modification budgétaire: la même opération que la précédente mais ce deuxième mouvement l'intervient que par une modification budgétaire.
- Enregistrement d'une modification budgétaire: le crédit mis à jour d'un article est augmenté ou diminué à concurrence du montant de la modification budgétaire. La date de la modification budgétaire et le montant doivent être enregistrés.
- Enregistrement d'un transfert: l'opération se déroule en plusieurs phases:
 - tester si l'article "de réserve" est suffisamment approvisionné, sans quoi le mouvement est rejeté;
 - soustraire le montant du transfert du crédit mis à jour de l'article "réserve";
 - ajouter le montant du transfert au crédit de l'article bénéficiaire du transfert;
 - garder le montant du transfert avec sa date de création.
- Suppression d'un article du budget: ce traitement consiste à inactiver un article dans le budget provincial. C'est une suppression logique (l'article continue à exister physiquement dans le fichier). Cette inactivation empêche tout mouvement sur l'article concerné.
- Suppression d'une modification budgétaire: opération consistant à soustraire le montant de la modification budgétaire du crédit mis à jour de l'article auquel se rapporte la modification considérée.

- suppression d'un transfert: il s'agit d'additionner le montant du transfert dans le crédit mis à jour de l'article "réserve". Ensuite, on soustrait ce montant du crédit mis à jour de l'article du fichier budget.

3.2.2. Mises à jour du budget analytique (5)

- Création d'un article: cette opération s'effectue en plusieurs phases: - tester si le disponible non engagé de l'article père est suffisant; si ce crédit est insuffisant, le mouvement est refusé par le système, sinon le système soustrait le montant de l'article analytique du disponible non ventilé de l'article père. Le montant de l'article lors de sa création doit toujours être gardé (pour faciliter le contrôle).

- Création d'un mouvement de transfert: un transfert ne peut être exécuté que si le disponible non ventilé de l'article père est suffisamment approvisionné; lorsque ce crédit est disponible, le système additionne le montant du transfert au crédit mis à jour de l'article analytique, ensuite, on soustrait le montant du transfert du disponible non ventilé en comptabilité analytique d'exploitation de l'article père.

- Suppression d'un article: c'est le transfert vers le disponible non ventilé en comptabilité analytique de l'article père réalisé à partir du disponible non engagé de l'article. Après ce transfert, l'article analytique est inactivé.

3.3. Editions des budgets

L'élaboration du budget provincial passe par plusieurs phases:

- prévisions budgétaires des établissements et services provinciaux;

- pré-budget modifié par la Commission des Finances;

- pré-budget modifié par la Députation Permanente;

- pré-budget modifié par la Commission des Finances du Conseil Provincial;

- budget modifié par le Ministère de l'Intérieur.

A chacune de ces phases, plusieurs documents sont édités en plusieurs exemplaires. Une fois les fichiers budget constitués, les gestionnaires doivent pouvoir les consulter.

Nous examinons ci-après quelques états imprimés produits par le système informatique actuel.

Ces états, classés par type de budget, ont l'inconvénient de fournir à l'utilisateur toutes les informations disponibles pour lui; or, il se peut que ce dernier ne s'intéresse qu'à une quantité limitée de ces informations.

3.3.1. Le budget provincial

Il est édité lors de chaque mise à jour et l'on distingue deux types d'imprimés: les récapitulatifs et l'ensemble du budget.

3.3.1.1. Etats imprimés

(voir le format à la page 28)

Ce schéma ne représente qu'une page de listing..Il sera le même pour le budget ordinaire, extraordinaire et pour ordre sauf que pour un schéma représentant le budget pour ordre, les colonnes "obligatoire" et "facultatif" seront remplacées par une seule colonne intitulée "indicatif".

Les recettes et les dépenses ne seront pas mélangées sur un même listing.

Type de mouvement signifie: modifications budgétaires ou transferts.

Un total sera édité par code économique et par fonction pour les types de mouvement, les crédits obligatoires et facultatifs.

3.3.1.2. Caractéristiques de l'édition (5)

- volume: pour un exercice budgétaire, le nombre d'articles du fichier budget provincial est: \pm 600;
- fréquence de mises à jour: quatre fois par an en principe, pour la phase de création (a), on effectue environ six mises à jour;
- délai de réponse: fourni par le système lors d'une demande d'édition:
 - 24 heures en période de création;
 - 48 heures pour les autres éditions.
- éditions des récapitulatifs: Des tableaux récapitulatifs faisant connaître les recettes et les dépenses pour une année, sont édités. Il existe trois types de tableaux récapitulatifs:
 - édition d'un récapitulatif par fonction pour les exercices antérieurs (format p.29).
.....
Un total de crédits obligatoires et facultatifs est effectué par fonction suivant la nature d'article;
 - édition d'un récapitulatif par fonction pour l'exercice propre (b) (format p.30)
.....

(a) création du budget.

(b) exercice en cours.

On voit que la présentation est différente de la précédente, il y a le code économique en plus et il faut effectuer le total pour toute l'année.

- édition d'un récapitulatif général par nature du budget
.....
(format p.31)

Les recettes et les dépenses se trouveront sur un même tableau récapitulatif.

Les caractéristiques de ces 3 tableaux sont les mêmes que celles du budget provincial (volume et fréquence des mises à jour).

3.2.2. Le budget analytique

L'édition du budget analytique est la production par le système d'un document unique en multiples exemplaires; un exemplaire est remis à chaque service (5).

3.3.2.1. Etat imprimé.
.....
(format d'édition p.32)

- Le crédit budgétaire global est le total des crédits alloués à une section plus le solde non ventilé en comptabilité en analytique.
- Article du budget: le code de l'article père au budget provincial.
- Numéro: c'est le numéro de transfert budgétaire.

3.3.2.2. Caractéristiques de l'Edition (5)
.....

- Volume: pour un exercice budgétaire, le nombre d'articles du fichier signalétique est d'environ 800 articles.
- Fréquence des mises à jour: en moyenne douze fois par an. L'édition se fait à chaque mise à jour.
- Délai de réponse: le système donne une réponse dans les 48 heures.

Conclusion
.....

Les états présentés ci-dessus éclairent certes et facilitent le choix que les autorités responsables de la décision doivent exercer. Mais pour le cas des décisions simples liées aux attributions plus ou moins courantes de chaque service, de nombreuses opérations de gestion seraient grandement allégées si les gestionnaires disposaient d'un moyen simple d'accéder aux informations; dans (12), on souhaite parvenir à "mettre à jour les fichiers selon une fréquence quotidienne et obtenir par exemple au jour le jour, la situation de différents articles du budgets ...".

Le système actuel, tel que présenté dans (5), comme nous venons de le décrire brièvement, ne répond pas encore à cette nécessité. Il existe des projets informatiques en cours de réalisation ou déjà réalisés à ce sujet; nous n'avons pas pu, malheureusement, les examiner.

PROVINCE DE NAMUR

B U D G E T

BUDGET nature du budget - nature de l'article - EXERCICE I9XX

FONCTION :

CODE ECONOMIQUE :

ARTICLE	LIBELLE	CREDIT INITIAL	TYPE DE MOUVEMENT	CREDIT MIS A JOUR	
				OBLIGATOIRE	FACULTATIF
-					

TOTAL CODE ECONOMIQUE : _____

TOTA L FONCTION : _____

RECAPITULATIF DES nature de l'article

EXERCICES ANTERIEURS

FONCTION	CREDITS OBLIGATOIRES	CREDITS FACULTATIFS	TOTAL

RECAPITULATIF DES nature de l'article

EXERCICE PROPRE

FONCTION	CODE ECONOMIQUE	CREDITS OBLIGATOIRES	CREDITS FACULTATIFS	TOTAL

TOTAL DES nature de l'article:
DE L'EXERCICE PROPRE

RECAPITULATIF GENERAL

DU

BUDGET nature du budget

	RECETTES	DEPENSES	SOLDE
EXERCICES ANTERIEURS :	_____	_____	_____
EXERCICE PROPRE :	_____	_____	_____
TOTAUX :	_____	_____	_____

PROVINCE DE NAMUR

BUDGET ANALYTIQUE DES ETABLISSEMENTS PROVINCIAUX

BUDGET nature du budget - nature de l'article - EXERCICE I9 XX

SERVICE XX : X(50)

ARTICLE DU BUDGET: X(II)

CREDIT BUDGETAIRE GLOBAL: 9(II)

SECTION XX: X(40)

NATURE DE CHARGE	CREDIT ANALYTIQUE INITIAL	NUMERO	TRANSFERTS	CREDIT ANALYTIQUE MIS A JOUR

TOTAL DE LA SECTION XX: _____

CREDIT ANALYTIQUE DU SERVICE XX: _____

SOLDE NON VENTILE EN COMPT. ANALYTIQUE: _____

CREDIT BUDGETAIRE GLOBAL : _____

3.4. Problèmes à résoudre

Les documents décrits au paragraphe précédent fournissent périodiquement des informations générales sur l'état des budgets provincial et analytique pour tous les services.

Ces états volumineux (plusieurs pages de listing) ne sont pas ^{nécessaires} lorsque l'utilisateur désire connaître des informations partielles et récentes sur le budget de son service.

En particulier, les temps de réponse souhaités sont très courts (moins de 24 heures) et les types de question susceptibles d'être posées sont très variés pour diverses raisons:

- Les responsables de gestion des services provinciaux n'étant pas intéressés par toutes les informations relatives à l'ensemble des articles des budgets, il est de la plus grande importance d'adapter la présentation et le volume de l'information à la fonction de celui qui la reçoit et de n'éditer que le minimum de textes imprimés.

- La Province étant divisée en plusieurs services et établissements correspondant à des affectations géographiques différentes, chaque service ou établissement travaille sur des informations nécessaires à ses travaux et différentes de celles d'autres secteurs. Cette décentralisation des tâches et dispersion géographique entraîne le besoin d'un système permettant à chacun d'avoir les seules informations qui lui paraissent utiles.

- Le responsable d'un centre d'activité provincial doit pouvoir contrôler la consommation de son budget par article budgétaire. Pour ce faire, il doit avoir à sa disposition des éléments d'informations valides indispensables à sa gestion, de façon à rendre compte dans le détail et immédiatement des conditions de la gestion. C'est au niveau du gestionnaire de base que se situe la connaissance exacte d'un service.

- Permettre au responsable de chaque établissement ou service provincial de suivre, tant en quantité qu'en valeur, le rythme de modifications (dépenses et recettes) du budget dont il a la charge, simplifiera la constitution des prévisions budgétaires par service.

3.5. Conclusion

Les utilisateurs ne désirant pas nécessairement disposer d'un seul coup des informations auxquelles ils peuvent accéder, il faut donc éviter de leur fournir des informations inutiles pour leur problème de l'heure.

Il est par conséquent utile de fournir à l'ensemble des utilisateurs, un jeu de programmes d'édition et des facilités leur permettant d'obtenir les éléments des budgets qui leur paraissent nécessaires.

Ces éléments doivent être présentés à l'utilisateur dans un format adéquat et compréhensible.

3.6. Solution proposée

A la lumière de ce qui précède, il apparaît que les gestionnaires éprouvent deux sortes de besoins pour mener à bien les travaux dont ils ont la charge.

Il faut que le système informatique permette:

- soit d'extraire un grand nombre d'informations des "fichiers budget" pour effectuer des travaux de type transfert, consultation, modifications budgétaires suivis d'édition;

- soit sélectionner rapidement un petit nombre d'informations qui satisfont aux critères d'une gestion posée.

Pour résoudre ces différents problèmes, nous proposons un langage de commande destiné aux utilisateurs non informaticiens. Les objectifs et les spécifications de ce langage sont décrits au chapitre suivant.

La nature des problèmes à résoudre (a) nous a conduit à proposer aux utilisateurs deux grandes classes des états de sortie:

-les formats standards: nous regroupons sous cette rubrique les types d'imprimés dont les formats ont été préétablis.

Ces états, nous venons de les décrire. Ils ont été dessinés par le responsable des finances provinciales; leur production par le système informatique nécessite l'accès à un grand nombre d'informations et beaucoup de traitements.

L'intérêt de tels formats pour notre système est d'épargner aux utilisateurs du langage proposé, le paramétrage de commandes des formats compliqués (exemple: édition du budget analytique, ce qui avait pour conséquence de rendre le langage beaucoup plus complexe.

Toutefois, le langage prévoira une commande pour l'obtention de ces états.

Les formats standards seront utilisés pour les grosses éditions existantes et d'autres qui pourraient être créées suivant les nécessités de gestion.

-les formats libres (b): utilisés lorsqu'un gestionnaire souhaite interroger les fichiers de différents budgets pour certains problèmes particuliers à son service. Vu le nombre des établissements provinciaux et la variété des questions susceptibles d'être posées, il devient impossible de fixer un format standard d'avance. D'où l'intérêt des formats libres, c'est à dire des formats plus simples, adaptés au type de problème posé et que l'utilisateur demande au système au moyen de commandes du langage.

(a) voir typologie des problèmes d'édition p. 53.

(b) voir les types de format libre au paragraphe 52.2 .

CHAPITRE IV: DESCRIPTION DU LANGAGE

=====

4.1. Généralités

4.1.1. Introduction

Les états imprimés décrits au chapitre précédent fournissent aux utilisateurs une quantité d'informations dont ces derniers ne se servent pas nécessairement.

Supposons qu'un utilisateur, à un instant précis, désire connaître le montant du crédit mis à jour pour un article du budget provincial. Il est tout à fait inutile de lui fournir un listing contenant tous les articles du budget provincial, car dans ce cas, il ne s'intéresse qu'à un seul article du budget.

Cette diffusion d'informations budgétaires devrait donc être adaptée aux besoins des gestionnaires afin qu'ils obtiennent uniquement les informations dont ils ont besoin. Par ce procédé, on éviterait de "noyer" les utilisateurs sous une marée d'informations.

Le langage que nous proposons ne se veut pas un langage de programmation. Nous souhaitons rester proches des conceptions de l'utilisateur qui n'est pas toujours informaticien. Pour cette raison, ce langage doit être simple, tant à comprendre qu'à employer. Il doit être défini en fonction de ce que l'utilisateur veut exprimer et avec l'intention de lui faciliter la tâche. Nous ne voulons pas y retrouver la complexité de certains langages évolués qui ne sont généralement utilisés que par les informaticiens.

Pour décrire le type d'action qu'il veut voir exécuter, l'utilisateur aura à sa disposition un certain nombre de commandes. Il s'agit en fait d'un ensemble de mots qui appellent chacun une fonction du système associée à une action bien déterminée.

Dans ce chapitre, nous décrivons le langage proposé.

4.1.2. Comparaison du langage avec les langages de consultation de données

Pour permettre aux gestionnaires et aux responsables du contrôle de gestion d'obtenir les renseignements qu'ils souhaitent, pourquoi ne pas utiliser un des langages de consultation existants, notamment ceux employés dans les systèmes de gestion de base de données?

Contrairement à ce que l'on pourrait être tenté de croire, ces langages ne sont pas toujours d'un emploi aisé pour un non informaticien.

Afin que le lecteur puisse se rendre compte des différences existant entre ces langages et le langage que nous proposons, nous avons examiné sommairement deux types de langages dans leur aspect "utilisation":

-le langage SOCRATE

-le langage IQF

4.1.2.1. Le langage SOCRATE

C'est un langage spécialisé, intimement lié au type de structure traitée par le système SOCRATE.(a).

(a) modèle de gestion de base de données réalisé par M. Abrial, à l'Université de Grenoble et développé par CII et ECA (8)

Socrate permet aux utilisateurs de réaliser un certain nombre de fonctions telles que:

- définition de la structure des informations;
- protection des informations stockées;
- mise à jour en batch ou en conversationnel avec validation des informations stockées;
- accès ponctuel aux informations;
- accès en séquence aux informations;
- suppression des objets de la base;
- recherche des objets de la base.

Les modifications, suppressions, recherches sont définies par un seul langage de manipulation:

-le langage de requêtes.

La citation permettant de définir l'occurrence recherchée est indépendante de la commande dont cette occurrence est l'objet.

C'est uniquement une suite de caractères qui servira à déterminer la fonction que l'on désire sur la base de données. Cette suite peut être réduite à un seul caractère. Exemple: soit une citation en SOCRATE.

"UNE PERSONNE AYANT NOM: DUPONT"

- Si on désire connaître cette personne, on écrira:

I UNE PERSONNE AYANT NOM: "DUPONT":?

- Pour la supprimer, on écrira:

S UNE PERSONNE AYANT NOM: "DUPONT"; ?

- Si on désire modifier la profession de cette personne, on aura:

M PROFESSION DE UNE PERSONNE AYANT NOM: "DUPONT",: "EMPLOYE" ?

Ainsi, pour remplir les différentes fonctions, il suffit d'ajouter au langage de "citation" une série de caractères qui déterminent ces ordres.

Une base de données SOCRATE est implantée sur disque en utilisant le concept de mémoire virtuelle: une technique de "haschcoding" permet de réaliser le passage mémoire virtuelle - mémoire réelle.

Comme on pourra le constater, nous nous sommes placés délibérément du point de vue de l'utilisateur sans chercher à entrer dans les détails du langage SOCRATE.

De ce qui précède, on peut dire:

- SOCRATE est un langage non dédié à une application particulière et est utilisé pour des bases de données volumineuses quant aux informations stockées.
- Ce langage, tout en ayant un aspect proche du français, est lourd de formulations, assez compliqué par sa syntaxe pour un non-informaticien et nécessite un apprentissage soigné. Le temps de formation est d'environ 1 mois (8).
- Le langage SOCRATE permet à l'utilisateur d'effectuer tous les traitements classiques de gestion.

4.1.2.2. L'IQF (Interactive Query Facility) (10)

C'est un langage employé par les utilisateurs d'une base de données en structure IMS (a), permettant sans programmation préalable d'accéder aux données par des opérations simples.

(a) Information Management System. C'est un système de gestion de base de données utilisé par IBM.

Exemples de commandes:

.....

- "QUERY COMMAND": - indique le nom de la base d'où seront sélectionnées les données.
 - Si fsalaire est le nom de la base de données, "SALAIRES", on peut écrire:
 - QUERY fsalaire
- LIST COMMAND: permet l'édition des données suivant un format prédéfini dans le système pour chaque donnée. Soit à éditer la liste des valeurs des caractéristiques, nom, salaire et département, en supposant que toutes ces données existent dans le système; on écrira:
 - LIST nom, salaire, département.

Toutes les commandes d'un langage IQF sont simples, car ce sont des formulations proches de l'anglais naturel. Mais c'est un langage peu destiné aux non-informaticiens (8), étant donné qu'il suppose une certaine connaissance de l'IMS, ce qui exige de bonnes notions d'informatique.

4.1.2.3. Conclusion

.....

Ces deux types de langage ne conviennent pas, à notre avis, pour résoudre le problème qui nous concerne et ce, pour diverses raisons:

- le langage SOCRATE comme le langage IQF est surtout un langage orienté vers les informaticiens. Il demande un effort soutenu d'apprentissage et une connaissance suffisante des modèles de bases de données utilisées (SOCRATE ou IMS), voire des notions de programmation (édition).

- Ces langages ne sont pas adaptés à une application particulière et ne sont utilisés que dans un environnement "base de données". L'utilisation ou le choix d'un système de gestion de base de données pose d'énormes problèmes d'investissement humain et matériel qui ne se justifie pas pour l'application que nous avons choisie.

Comme nous l'avons annoncé dans l'introduction, le langage que nous proposons doit être d'un apprentissage très facile pour les non-informaticiens. Il peut être utilisé avec les fichiers classiques et est conçu pour résoudre des problèmes de gestion relatifs aux budgets provinciaux.

4.1.3. Objectifs du langage

Mettre à la disposition des gestionnaires des finances provinciales un outil leur permettant de sélectionner à la demande, les éléments d'informations nécessaires à leur gestion.

- Le langage doit permettre 3 possibilités de sortie:
 - sortie sur imprimante;
 - sortie sur bande magnétique;
 - sortie sur terminal.
- Le langage est interactif et apporte une aide à l'utilisateur dans la formulation de sa requête.

En cas d'erreur dans l'écriture d'une commande par l'utilisateur, il est souhaitable que le système envoie un message compréhensible par l'utilisateur, pour lui demander éventuellement de reprendre la commande.

4.2. Eléments nécessaires à une édition

4.2.1. Questions susceptibles d'être posées par l'utilisateur

Nous allons examiner un ensemble de questions susceptibles d'être posées sur les différents budgets. Nous ne pouvons évidemment pas citer la totalité de ces questions.

Les informations demandées varient d'un service à l'autre et sont fonction du temps (périodicité de certaines transactions, intérêt naissant pour de nouvelles requêtes, changements imprévus quant à la réglementation de la gestion).

Le langage doit être conçu pour répondre à un maximum de requêtes.

4.2.1.1. Questions relatives au budget provincial

- Etant donné un exercice budgétaire, éditer le budget et le récapitulatif du budget par nature du budget, par nature d'article.
- Etant donné une nature de budget, éditer le budget et le récapitulatif du budget par nature d'article.
- Pour une nature d'article, éditer le récapitulatif.
- Pour une fonction, éditer le total des crédits par code économique
- Connaissant un article du budget, éditer le crédit initial, le crédit mis à jour, les modifications budgétaires, les transferts, le solde non ventilé.
- Editer le crédit initial et le crédit mis à jour.
- Réaliser les modifications budgétaires et/ou les transferts suivis d'édition.
- Editer les articles concernés par une modification.

4.2.1.2. Questions relatives au budget analytique
.....

- Connaissant un exercice budgétaire, quels sont les totaux de crédits de la nature de charge ?

- Etant donné un article du budget analytique, quel est son crédit initial, son crédit mis à jour, ses transferts ?

- Etant donné un service, éditer le total de crédits par nature de charges.

- Pour une nature de charge, quel est le total de crédits par service, par section?

4.2.2. Analyse de la forme des questions

Pour répondre à toutes ces questions, la démarche à suivre est toujours la même:

- rechercher les objets qui satisfont à une condition donnée;
- rechercher certaines informations sur ces objets;
- opérer certains traitements élémentaires sur des informations pour obtenir de nouvelles informations ne figurant pas dans les fichiers;
- afficher les résultats sur un support de sortie.

Nous allons le constater en analysant quelques questions.

4.2.2.1. Description de quelques questions
.....

Soit la question:

- "Etant donné un exercice budgétaire, éditer le budget et le récapitulatif du budget par nature du budget, par nature d'article".

Pour que le système puisse fournir une réponse à cette ques-

tion de l'utilisateur, les paramètres utilisés dans les commandes du langage doivent fournir des indications qui permettent l'identification du fichier nécessaire à l'affichage des résultats.

Dans le cas présent, il s'agit du budget provincial.

Le choix du fichier étant fait, le système procède à une sélection de données. Dans la question posée, plusieurs sélections sont possibles:

- sélection par nature du budget (il existe 3 natures du budget);
- sélection par nature de l'article (il existe 2 natures d'articles.

Après la sélection des données nécessaires, le système procède à l'édition. Il faut naturellement fournir des indications sur la nature de support de sortie et le format de sortie.

Pour nous résumer, les commandes du langage doivent avoir la possibilité de préciser au système à quel sous-ensemble ou à quel élément d'un fichier on s'intéresse. Tout ceci est complètement ignoré de l'utilisateur. Des critères de sélection sont nécessaires pour isoler les seules données intéressantes. Si nous reprenons la question, pour un même budget, il peut arriver, et ce sera souvent le cas dans la pratique, que l'utilisateur ne s'intéresse qu'à une nature de budget et une nature d'article.

En réponse à cette question, le système peut fournir plusieurs états. Les réponses possibles sont les suivantes:

- le budget ordinaire - recettes
- le budget ordinaire - dépenses
- le budget extraordinaire - recettes
- le budget extraordinaire - dépenses
- le budget pour ordre - dépenses

- le budget pour ordre - recettes
- le récapitulatif des recettes par article du budget ordinaire;
- le récapitulatif des dépenses par article du budget ordinaire;
- le récapitulatif des recettes par article du budget extraordinaire;
- le récapitulatif des dépenses par article du budget extraordinaire;
- le récapitulatif des recettes par article du budget pour ordre;
- le récapitulatif des dépenses par article du budget pour ordre.

Comme il est peu probable qu'un utilisateur demande d'un seul tenant tous ces états, il faut que le langage possède un critère lui permettant d'effectuer un choix parmi tous les états possibles.

Poursuivons l'analyse en examinant une seconde question:

- "pour une fonction, éditer le total des crédits par code économique".

La démarche à suivre par le système sera la suivante:

- identification du fichier "budget provincial";
- identification du fichier "fonction";
- sélection de la fonction demandée;
- sélection de tous les codes économiques appartenant à cette fonction;
- sélection des articles par code économique;
- édition des totaux des crédits par code économique;
- totalisation des crédits mis à jour des articles appartenant à un même code économique.

l'ensemble ci-dessus montre que le langage proposé doit permettre à l'utilisateur non seulement de choisir et de spécifier les données qui figurent dans les fichiers et qu'il désire voir éditées, mais aussi de créer de nouvelles données au moyen d'opérateurs appropriés. Dans notre cas, il faut un opérateur de totalisation.

Soit la question suivante:

- "Connaissant un article du budget, éditer le crédit initial, le crédit mis à jour, les modifications budgétaires, les transferts, le solde non ventilé".

La démarche sera la suivante:

- choix du fichier budget provincial;
- sélection de l'article à partir d'un critère fourni par l'utilisateur au moyen d'une commande;
- identification et sélection des éléments demandés:
 - crédit initial;
 - crédit mis à jour;
 - modifications;
 - transferts;
 - solde non ventilé;
 - édition sur un support de l'ensemble des éléments demandés.

Cette question ne nécessitant aucune opération sur les données, celles-ci seront éditées après leur sélection, dans un journal d'édition adéquat.

- "Réaliser les modifications budgétaires suivies d'une édition".

La démarche sera la suivante:

- sélectionner les articles à modifier;

- opérer les modifications nécessaires;
- écrire les articles modifiés dans les fiches;
- éditer le ou les articles modifiés sur un support de sortie.

On remarque dans cet exemple, la nécessité de permettre à l'utilisateur d'effectuer les modifications budgétaires suivies d'une édition. Un opérateur de modifications doit être mis à sa disposition.

-" Réaliser les transferts suivis d'une édition".

La démarche est analogue à la précédente:

- sélectionner l'article "origine" du transfert;
- sélectionner l'article "destination" du transfert;
- opérer le transfert;
- éditer les articles concernés par les transferts (sur un support de sortie).

On remarque la nécessité de mettre à la disposition de l'utilisateur un opérateur de transfert.

-"Quels sont les totaux des crédits d'une nature de charge donnée, connaissant un exercice budgétaire"?

Le système procède comme suit:

- sélection du fichier budget analytique;
- sélection de la nature de charges pour l'exercice demandé;
- totalisation des crédits mis à jour pour tous les articles analytiques ayant cette nature de charges;
- édition des totaux demandés.

On remarque ici que l'édition demandée doit être précédée par une opération de totalisation. Comme pour les autres questions, il apparaît clairement dans cet exemple, que l'édition doit être précédée par une phase de sélection qui

décrit l'ensemble des réalisations de l'objet auquel on s'intéresse. Cette sélection permet d'"isoler" les données demandées.

- "Etant donné un service, éditer le total des crédits pour une nature de charges".

La demande est la suivante:

- sélectionner le fichier budget analytique;
- sélection du service (le code service ou le nom du service peut servir de critère de tri);
- sélectionner pour ce service les articles ayant la nature de charges donnée;
- effectuer la totalisation des crédits mis à jour pour ces articles;
- éditer le total demandé.

On remarque encore ici l'apparition de l'opérateur de totalisation, cet opérateur doit précéder l'édition demandée.

- "Pour une nature de charges, quel est le total des crédits par service, par section"?

La procédure à suivre respecte la même démarche:

- sélectionner le fichier "budget analytique";
- rechercher le service;
- rechercher toutes les sections qui appartiennent à ce service et ayant une même nature de charge;
- effectuer le total de crédits de toutes les sections appartenant à ce service pour constituer un total de service;
- éditer le total des crédits pour ce service;
- recommencer la même démarche pour chaque service.

Cet exemple montre un cas où la totalisation se fait à trois niveaux différents:

- une opération de totalisation se situe

d'abord au niveau des crédits appartenant aux articles d'une même section pour constituer un total de section;

ensuite, les sous-totaux des sections sont cumulés à leur tour pour constituer un total de service;

les totaux des services peuvent à leur tour être additionnés pour obtenir le total général des crédits du budget analytique.

4.2.2.2. Méthode de traitements des questions

On peut effectuer une analyse identique pour toutes les autres questions.

L'examen des questions susceptibles d'être posées et la diversité des applications montre de façon évidente qu'il est impossible, à un moment donné, de dresser une liste exhaustive des questions à traiter.

Suivant les réglementations de gestion et le degré de responsabilité des gestionnaires, les informations dont a besoin un utilisateur sont très variables. Au niveau de la Députation Permanente, les questions sont de savoir si oui ou non, tous les centres de frais en général ou un établissement provincial, ont respecté les normes de gestion fixées.

Les responsables des centres de frais par contre demandent dans le plus grand détail, les éléments qui dépendent de leur gestion (budget qui leur est alloué en début d'année et le rythme de consommation de celui-ci).

Pour un même niveau de responsabilité, la diversité des applications implique une diversité de questions.

Nous avons dès lors pris comme option de ne pas chercher à répondre uniquement à une liste exhaustive de questions.

Ceci permet à l'utilisateur la possibilité de poser toutes les questions faites dans le paragraphe précédent.

Si l'on se réfère à cette analyse;

- sélection des objets qui satisfont un critère de choix;
- recherche de certaines caractéristiques relatives aux objets sélectionner,

On constate que les conditions données peuvent s'exprimer en fonction d'une série de conditions élémentaires.

Exemple: éditer le crédit mis à jour de l'article du budget ordinaire dont l'exercice est Y.

L'identifiant de l'article est Y.

Sous forme de relation la condition de sélection de cet objet peut se formuler de la manière suivante:

IDENTIFIANT= Y ET BUDGET= ordinaire ET EXERCICE = Y.

Une fois qu'il a déterminé l'objet qui satisfait cette condition, le système peut rechercher les informations relatives à cet objet.

En conclusion, nous pouvons dire que la liste des questions que l'on peut traiter n'est pas exhaustive, mais le champs en est limité par la liste des relations de base que nous examinerons plus loin.

4.2.3. Conclusion de la description des questions

4.2.3.1. Opérateurs à mettre à la disposition de l'utilisateur

L'analyse identique de différentes questions a conduit à la nécessité de mettre à la disposition de l'utilisateur un certain nombre d'opérateurs qui deviendront les commandes du langage. Chaque opérateur prendra pour données de base les objets figurant dans le sous-schéma budgets que nous avons décrit.

Dans le stade actuel de l'étude, les opérateurs suivants ont été jugés nécessaires:

- un opérateur de sélection: permet de repérer et de sélectionner les objets qui répondent à des conditions portant sur une ou plusieurs de leurs caractéristiques.
- opérateur de totalisation: permet d'obtenir des données globalisées qui ne figurent pas comme telles dans les fichiers.
- opérateur de transfert: permet d'effectuer les mouvements de transferts.
- opérateur de modification: permet d'effectuer les modifications budgétaires.
- opération d'édition: nécessaire pour permettre l'édition des nombres.

Ces opérateurs sont des fonctions primitives qui permettent d'exprimer les requêtes des utilisateurs.

4.2.3.2. Paramètres à faire figurer dans une commande

- paramètre de désignation de données: il faut indiquer les articles et les données qui sont utilisées dans une requête de l'utilisateur. Ceci permettra de déterminer la structure de données qui devra être prise en considération par un opérateur. C'est sur base de ces renseignements qu'il sera possible de répondre aux questions des utilisateurs.
- paramètres exprimant les critères de sélection: ces paramètres permettent à l'utilisateur d'exprimer des conditions de sélection d'objets. Ces conditions permettent au système d'effectuer des sélections et des extractions sur certains articles répondant aux conditions données. Ces paramètres sont nécessaires pour l'opération de sélection.

- paramètres de spécification du support de sortie: il faut prévoir un paramètre de spécification du support de sortie. Les données peuvent être fournies directement à l'utilisateur sur 2 types de support:

- soit le terminal: ce support est utile quand la quantité d'informations à éditer est assez réduite;

- soit l'imprimante: ce support est utilisé pour les éditions qui impliquent des volumes d'informations assez élevés.

- paramètre de spécification du format de sortie: la nature du support de sortie et la forme de présentation des états souhaités par l'utilisateur impose un choix de format:

- sortie de données sous forme compacte;
- édition des titres, des en-têtes de pages;
- édition des totaux, etc

La nature de l'application et les conditions d'exploitation du système imposeront aux utilisateurs un choix de format de sortie parmi les formats qui seront à leur disposition.

CHAPITRE V: TYPOLOGIE DES EDITIONS

=====

Avant de déterminer les types de formats qui seront mis à la disposition de l'utilisateur, nous avons examiné un certain nombre de critères qui caractérisent l'édition des informations à éditer. C'est la mise en commun de ces critères qui nous a conduit à proposer quelques formats pour l'utilisateur.

5.1. Critères de classification des états imprimés

5.1.1. Volume des informations à éditer

Par volume, nous entendons la quantité d'informations en nombre de lignes ou pages de listing. On peut imprimer des états ne dépassant pas quelques lignes de listing. Ainsi, si un utilisateur désire connaître le solde non ventilé d'un article budgétaire, cette édition ne comportera qu'une ligne imprimée au terminal.

Par contre, certaines éditions nécessitent plusieurs pages de listing. C'est le cas notamment lorsqu'il faut éditer tout le budget provincial.

Le volume des informations à traiter aura une influence sur le choix du support de sortie. Il est évident que les grosses éditions seront sorties à l'imprimante; les autres seront affichées soit au terminal, soit à l'imprimante.

5.1.2 Temps d'attente souhaité par l'utilisateur

Lorsque l'utilisateur pose des questions ponctuelles au système, les temps de réponse acceptables sont assez réduits. Ces questions ne concernent souvent qu'un article du budget et les gestionnaires désirent une réponse dans les plus brefs délais. Pour ce type d'édition, la mise en page de l'état n'est

pas compliquée à réaliser et il faut un traitement en mode interactif.

Un gestionnaire qui désire connaître la date d'une modification budgétaire ne devrait pas attendre 10 heures pour l'obtenir!

5.1.3. Degré de complexité de la mise en pages

Les états imprimés nécessitent pour être édités des descriptions de mise en forme plus ou moins compliquées suivant la nature de l'état demandé. Les principales caractéristiques qui entrent en ligne de compte sont les suivantes:

- ruptures de page : certains états imprimés nécessitent des sauts de page après un certain nombre de lignes ou un critère choisi. Ainsi, dans les états imprimés que nous avons analysé au chapitre III, pour l'édition du budget ordinaire, les utilisateurs exigent un changement de page après chaque changement de nature de budget, de manière à avoir les dépenses et les recettes toujours sur des pages séparées.
- introduction dans les états (a) des éléments calculés: certains états nécessitent l'impression de totaux généraux ou partiels. Nous avons déjà eu l'occasion de donner quelques exemples pour ce type de problèmes.
- description du contenu des lignes:
 - le nombre de titres par état: l'utilisateur peut demander l'impression de plusieurs titres pour un même état; exemple: l'édition du budget analytique (voir format présenté au chapitre III);
 - titres de pages: pour la clarté de l'état imprimé, on peut décider la répétition du titre de page à chaque page de listing;

(a) mis pour état imprimé

- en-têtes de colonnes: ce sont des sous-titres dans une page, ils peuvent soit être reproduits uniquement à la première page, soit être reproduits à toutes les pages;

- ligne des caractéristiques à éditer: (ligne détail en cobol); le langage d'édition doit permettre d'indiquer de façon simple les caractéristiques des objets qui doivent être édités.

5.1.4. Caractère cyclique de l'édition demandée

Les éditions à caractère cyclique sont celles qui sont demandées à une période déterminée à l'avance. Ces éditions se concrétisent par une importante production de documents standardisés. Nous pourrions classer dans cette catégorie les formats officiels d'édition qui sont actuellement en cours d'élaboration à la province. Ces documents concernent des volumes importants de données que l'on a l'habitude d'organiser préalablement.

Le format d'édition ne sera pas le même selon que l'on sera en présence d'une édition à caractère cyclique ou d'une édition à caractère exceptionnel.

Dans le premier cas, on connaît à l'avance le type d'informations demandées par les utilisateurs; dans le deuxième cas, par contre, on ignore complètement ce que demandera l'utilisateur.

5.2. Formats d'édition

La diversité des applications relatives aux éditions observées notamment lors de l'analyse des questions de l'utilisateur, les contraintes dues à la nature des supports de sortie et aux difficultés qu'éprouverait l'utilisateur à décrire les formats, nous ont amenés à proposer des formats par catégorie de problème.

Pour cela, nous avons groupé les formats d'édition en deux grandes classes, les formats standards et les formats libres.

5.2.1. Les formats standards

Les formats standards comprennent tous les formats d'édition présentés au chapitre III.

Pour rappel, il s'agit des formats utilisés pour l'édition du budget provincial, des différents récapitulatifs du budget provincial, du budget analytique des établissements provinciaux.

Ces formats ont été demandés et dessinés par les utilisateurs. Ils sont décrits dans le document "Système Informatique de la Province de Namur" (5).

Il n'est donc pas question pour nous de les modifier.

En reprenant les critères décrits ci-dessus, examinons ce qui définit ce type de format.

- Volume des informations: il est caractérisé par le nombre de pages du listing.

Lors de la présentation de ces formats, nous avons donné une estimation du volume en nombre d'articles. Le budget provincial compte en moyenne 800 articles pour une année budgétaire.

A raison d'un article par ligne, et sans tenir compte des titres, l'édition complète du budget demande 800 lignes d'impression.

- Le temps d'attente: pour ce type d'édition, le temps d'attente pour l'utilisateur est relativement long. Il est de l'ordre de quelques heures.

- La mise en page de ce type de format est assez complexe.

Pour une même liste, il est possible d'avoir plusieurs en-têtes d'état et plusieurs titres de page.

Il est également possible d'insérer dans une édition, certaines données "résultats": cumuls ou totaux intermédiaires.

Les sorties de ce type standard ont un caractère périodique. On ne les imprime pas quotidiennement.

En conclusion, vu le volume élevé des informations à éditer de ce type de format, il en résulte une édition à l'imprimante.

Dans le langage que nous proposons, nous n'avons pas prévu de commande spéciale. Suivant le système d'exploitation dont on dispose, l'utilisateur qui désire obtenir une sortie d'informations sous un tel format peut appeler des modules d'édition (a) appropriés à partir du terminal.

5.2.2. Les formats libres

Ces formats ont pour but de permettre à l'utilisateur "une gestion par explication", c'est à dire éditer parmi un ensemble de données, celles qu'il juge utiles, à un instant donné. L'étude des questions de l'utilisateur a fait apparaître

(a) Des modules d'édition doivent exister dans le système pour les différents imprimés.

tre de façon évidente l'intérêt de ces formats.

Nous avons proposé trois formats libres. Cette option a été décidée suite à la diversité des questions possibles et compte tenu des critères que nous nous sommes fixés.

5.2.2.1. Le mode 1

- volume des informations: du point de vue volume des informations, les états décrits sur ce mode peuvent comporter plus d'une page de listing, mais le volume de données produit reste inférieur au volume produit par les formats standards. Ce type de format est utile pour l'impression du budget relatif à un service provincial, alors que les états standards fournissent des états qui concernent tous les services provinciaux.

- Mise en page: -un ou plusieurs en-têtes d'état sont fournis par l'utilisateur sous forme de paramètres dans une commande (voir les commandes plus loin).

-rupture de page automatique.

- Support de sortie: imprimante, spécifié au moyen d'un paramètre passé par l'utilisateur; il est également possible d'inclure des données calculées parmi les données éditées.

5.2.2.2. Le mode 2

Le but de ce format est de permettre une meilleure adaptation de l'édition des informations aux désirs de l'utilisateur.

Il permet au gestionnaire de ne sortir que les informations qui l'intéressent à un moment précis.

C'est ainsi qu'un chef de service, loin de s'intéresser à tout le budget de son service, peut s'intéresser à une fonction (a) particulière de ce service ou à un code économique.

(a) "fonction" est utilisé ici dans le sens de rubrique comptable de regroupement.

Examinons les caractères de ce format.

- Le volume des informations sera de une ou quelques pages de listing.
- La mise en page se présentera comme suit:
 - il n'y a pas de rupture de l'en-tête d'état;
 - une rupture de page est possible mais elle est assurée automatiquement par le système.
 - les en-têtes de colonnes ne sont pas fournis par l'utilisateur comme paramètres mais le système va les générer automatiquement à partir de données à éditer.

Ce mode est utilisé pour les travaux au niveau d'un centre d'activité, lorsque, tout en ne s'intéressant qu'à une masse assez réduite de données, les gestionnaires souhaitent quand même une certaine présentation sur le listing.

5.2.2.3. Le mode 3

Ce format sera utilisé pour des informations de type question-réponse élémentaire, c'est à dire pour les nécessités d'une consultation rapide, ne demandant pas une mise en page élaborée. Ce mode sera généralement utilisé pour les petites éditions, consécutives à une mise à jour lorsque l'utilisateur désire s'assurer du succès de son opération.

Ce mode sera également employé lorsque l'utilisateur souhaite des renseignements sur l'état d'un élément de l'article budgétaire.

Dans tous les cas, il est inutile d'imprimer des titres d'état ou de pages qui feraient sans doute perdre quelques secondes au gestionnaire qui attend, au terminal, une réponse à la question ponctuelle qu'il a posée.

Les caractéristiques de ce mode général sont les suivantes:

- un volume des informations faible, ne dépassant pas une page de listing;
- un en-tête d'état inexistant;
- un en-tête de page inexistant;
- les en-têtes de colonnes sont générés automatiquement par le système, l'utilisateur ne les décrit pas;
- les totaux ne peuvent pas être édités dans ce mode;
- dans ce mode de format, l'édition se fera toujours au terminal.

5.2.2.4. Etats imprimés en formats libres: quelques exemples

Dans ce paragraphe, nous présentons quelques exemples de formats en mode 1, mode 2 et mode 3.

Les schémas qui suivent représentent une page de listing telle qu'elle doit être fournie par l'ordinateur à l'utilisateur, dans le système que nous proposons.

Symboles utilisés:

E= ligne en-tête d'état
P= ligne titre de page
C= ligne en-tête de colonne
D= ligne détail
R= ligne d'une "donnée résultat".

- Formats en mode 1: fig. F 1 1, F 1 2
- Formats en mode 2: fig. F 2 1, F 2 2, F 2 3, F 2 4
- Formats en mode 3: fig. F 3 1, F 3 2, F 3 3, F 3 4.

Pour une fonction, édition du crédit initial et mis à jour.

E	BUDGET ORDINAIRE		19 X X	
P	DEPENSES			
C	FONCTION	ARTICLE	CREDIT INITIAL	CREDIT MIS A JOUR
D	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____
R	TOTAL CREDIT INITIAL			_____
	TOTAL CREDIT MIS A JOUR			_____

fig. 1 1.

Edition du crédit total alloué à un code économique pour le budget extraordinaire de l'exercice 19XX.

E	BUDGET EXTRAORDINAIRE 19XX			
P	RECETTES			
C	FONCTION	CODE ECONOMIQUE	ARTICLE	CREDIT INITIAL
	_____	_____	_____	_____
R	TOTAL CODE ECONOMIQUE:			_____

fig. F 1 2.

Edition du crédit mis à jour total pour un service.

P	BUDGET ANALYTIQUE		
C	SERVICE	SECTION	CR2DIT MIS A JOUR
D	_____	_____	_____
R	TOTAL SECTION:		_____
	TOTAL SERVICE:		_____

fig. F 2 1

Edition du crédit total par section, par service pour une nature de charge.

P	BUDGET ANALYTIQUE			
C	NATURE DE CHARGE	SERVICE	SECTION	CREDIT
D	_____	_____	_____	_____
D		_____	_____	_____
		_____	_____	_____
		_____	_____	_____
R	TOTAL SECTION:			_____
	TOTAL SERVICE:			_____

fig. 2 2.

Edition de totaux des crédits d'une nature de charge pour l'exercice 19XX.

P	BUDGET ANALYTIQUE 19XX	
C	NATURE DE CHARGE	CREDIT
D	_____	_____

R		TOTAL CREDIT: _____

fig. F 2 3.

Edition du crédit initial, du crédit mis à jour des transferts du budget analytique pour un article.

P	BUDGET ANALYTIQUE			
C	ARTICLE	CREDIT INITIAL	CREDIT MIS A JOUR	TRANSFERTS
D	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____

fig. F 2 4

Edition des modifications budgétaires ayant influencé un article du budget.

C	ARTICLE	NUMERO (a)	MONTANT	DATE
D	_____	_____	_____	_____
	_____	_____	_____	_____
	_____	_____	_____	_____

fig. F.3 1.

Edition du crédit initial et du crédit mis à jour d'un article.

C	ARTICLE	CREDIT INITIAL	CREDIT MIS A JOUR
D	_____	_____	_____
	_____	_____	_____
	_____	_____	_____

Fig. F.3 2.

Edition du solde non ventilé d'un article.

C	ARTICLE	SOLDE NON VENTILE
D	_____	_____

fig. F 3 3.

(a) numéro de la modification.

Edition des articles concernés par une modification budgétaire.

C	NUMERO MODIFICATION	ARTICLE
D	_____	_____

fig. F 3 4.

CHAPITRE VI: PROPOSITION DES SPECIFICATIONS DU LANGAGE

=====

6.1. Notations formelles

Nous décrivons les commandes du langage en utilisant les notations formelles du langage Cobol. Nous en rappelons les principales ici:

- un mot-clé du langage est écrit en majuscules;
- un mot employé par l'utilisateur, en minuscules;
- un contenu facultatif est mis entre crochets;
- les alternatives sont regroupées entre { }
ou séparées par un trait vertical;
- une répétition est représentée par des points de suspension: ...

6.2. Les commandes du langage

Le langage que nous proposons met à la disposition de l'utilisateur non informaticien un certain nombre de commandes paramétrables. La plupart des commandes correspondent aux opérations décrites au paragraphe 4.2.3.1.

La liste de ces commandes ainsi que les opérateurs auxquels elles correspondent, est reprise dans le tableau suivant:

opérateur	commande
sélection	demande
totamisation	total
transfert	transfert
modification	modification
édition	affiche

6.2.1. La commande DEMANDE

Cette commande a pour but de sélectionner les données utiles pour les traitements demandés par l'utilisateur.

Elle précède toujours les autres commandes.

Nous avons volontairement séparé cette commande de celle d'édition (a), parce que certaines données, après avoir été extraites des fichiers, doivent subir certaines opérations (totalisations, modifications) avant d'être imprimées.

La formulation de la commande est la suivante:

DEMANDE valeur SI critère - de - - choix.

Dans cette commande, valeur représente la caractéristique de l'élément qui doit remplir certaines conditions pour être sélectionné.

SI est un séparateur qui introduit les critères de sélection.

Le critère - de - choix peut être décrit comme suit dans le langage de Bacchus (b):

$\langle \text{critère de choix} \rangle : : = \langle \text{critère arithmétique} \rangle [\text{ET} \langle \text{critère de choix} \rangle]$

$\langle \text{critère arithmétique} \rangle : : = \langle \text{expression arithmétique} \rangle \langle \text{opérateur logique} \rangle \langle \text{expression arithmétique} \rangle$

$\langle \text{expression arithmétique} \rangle : : = \left\{ \begin{array}{l} \langle \text{caractéristique} \rangle \\ \langle \text{constante} \rangle \end{array} \right\}$

$\langle \text{opérateur logique} \rangle : : = \langle / \leq / = / \neq / \rangle$

$\langle \text{expression arithmétique} \rangle : : = \langle \text{caractéristique} \rangle$

Les conditions à remplir peuvent prendre plusieurs formes. Voyons en quelques exemples.

(a) voir commande AFFICHE.

(b) il est entendu que nous utilisons cette notation pour des raisons de clarté du texte; pour un utilisateur, il faudrait peut-être utiliser moins de symboles et beaucoup de mots.

Pour une fonction, éditer le total des crédits par code économique.

Question: La commande de sélection peut s'écrire ainsi:

DEMANDE fonction SI code - fonction = 9 (3)

Lors de l'exécution de cette commande, le système va sélectionner une fonction dont le code fonction a été donné par l'utilisateur. L'édition proprement dite est réalisée par une commande d'édition.

Certains cas peuvent comporter des chaînes de conditions, c'est à dire plusieurs conditions liées par l'opérateur ET.

Exemples: - éditer le budget et le récapitulatif du Budget par nature du budget et par nature d'article pour l'exercice budgétaire X.

La sélection se formule comme suit:

DEMANDE article SI exercice = X et nature du

le budget (a) = ordinaire ET nature d'article = dépenses (a).

- Editer le récapitulatif des dépenses:

la sélection des articles est demandée comme suit:

DEMANDE article SI nature d'articles = dépenses.

Pour le système, lors de l'analyse de cette commande, chaque élément de la structure de l'article concerné est examiné.

Toutes les caractéristiques de l'article sélectionné par la condition sont validées pour les opérations d'édition. De même tous les articles situés à un niveau supérieur ou inférieur mais qui sont liés à l'article sélectionné, sont validés dans

(a) nature de budget et nature d'articles sont écrits en toutes lettres mais dans la pratique, il sera préférable d'utiliser des abréviations ou des codes pour diminuer les risques d'erreur dans l'écriture.

les mêmes conditions.

En conclusion, la commande DEMANDE n'effectue pas d'édition, elle sélectionne uniquement les données qui deviennent disponibles pour les autres opérateurs.

6.2.2. La commande total:

La syntaxe s'écrit:

TOTAL donnée - 1, donnée - 2 , donnée - 3 PAR donnée - 4

Cette commande permettra à l'utilisateur d'indiquer les données dont les valeurs doivent être sommées et éditées.

Donnée - 1, donnée - 2, donnée - 3 représentent 3 niveaux de rupture différents, l'ordre de grandeur allant croissant de gauche à droite.

Donnée - 4 est l'identificateur de la donnée élémentaire dont les valeurs successives seront cumulées pour constituer des sous-totaux. de niveau 1 représentés par donnée - 1. En d'autres termes, tout changement de valeur d'une donnée de référence donnée - 1 constituera une rupture de niveau 1. Juste avant une rupture, un sous-total sera imprimé.

Les sous-totaux intermédiaires de niveau - 1 seront à leur tour, sommés par groupes déterminés en fonction de la donnée de référence donnée - 2. Tout changement de cette donnée provoquera l'impression d'un sous-total de niveau - 2.

Les résultats intermédiaires (sous-totaux obtenus au niveau 2) peuvent à leur tour être cumulés avec la fonction des variations de la donnée de référence donnée - 3.

Nous nous sommes limités à 3 niveaux de rupture. L'utilisateur pourra s'arrêter au premier niveau mais il ne pourra jamais paramétrer plus de 3 niveaux de rupture, le niveau 3 indiquant

le total général (a).

Tous les sous-totaux de niveau 1 et 2 seront édités.

Exemples d'utilisation de la commande TOTAL:

- éditer le total des crédits mis à jour par service et par section pour une nature de charge X:
TOTAL section, service, PAR crédit mis à jour.
- éditer le total des crédits par code économique et par fonction:
TOTAL code économique, fonction PAR crédit budgétaire
- éditer le total des crédits par nature de charge:
TOTAL nature de charge, service PAR crédits analytiques.

6.2.3. La commande MODIF

Cette commande sera utilisée pour l'enregistrement d'une modification budgétaire. A l'issue de toute modification, une édition sera faite.

La syntaxe de cette commande est la suivante:

MODIF article AVEC numéro, montant , libellé , date.

- "article" désigne l'article qui doit subir la modification budgétaire et qui doit avoir été sélectionné par la commande DEMANDE;
- "AVEC" est utilisé pour des raisons de facilité de compréhension pour l'utilisateur;
- "numéro" sera le numéro de la modification budgétaire que l'utilisateur désire faire enregistrer;
- "montant" est le montant du crédit accordé en augmentation ou

(a): nous nous sommes arrêtés à 3 niveaux de ~~utilisation~~ **totalisation** pour des raisons de simplification du système.

en diminution du crédit disponible. C'est un nombre qui pourra être positif ou négatif selon qu'il s'agit d'une augmentation ou d'une diminution de crédit.

- "montant" est facultatif lorsque le crédit porte sur le libellé de l'article (a).

- "libellé": est le nouveau libellé de l'article. Il remplace le libellé actuel. Cette clause n'est pas utilisée lorsque la clause "montant" est employée.

- "date" est la date de la modification budgétaire.

6.2.4. La commande TRANSF

Cette commande est utilisée par les gestionnaires pour effectuer des opérations de transfert entre deux articles budgétaires. Les deux articles doivent déjà avoir été sélectionnés par deux commandes "DEMANDE".

Syntaxe de la commande:

TRANSF article - 1, article - 2 AVEC numéro, montant, date.

- "article - 1" est l'article origine du mouvement de transfert;
- "article - 2" est l'article de destination du transfert;
- "AVEC": mot réservé du langage, employé pour faciliter la compréhension de la commande par l'utilisateur;
- "numéro" est le numéro de transfert (b);
- "montant": le montant du transfert;
- "date": la date à laquelle a eu lieu le transfert

Tout mouvement de transfert sera suivi d'une édition.

(a): Pour rappel, on a vu que la modification budgétaire portait soit sur le montant du crédit, soit sur le libellé de l'article.

(b): doit être enregistré.

6.2.5. La commande "AFFICHE"

En formats libres, étant donné que les éléments définissant la constitution de l'état imprimé sont introduits par l'utilisateur, il est nécessaire de mettre à sa disposition une commande qui lui permette de réaliser une certaine mise en page de données de sortie.

AFFICHE commande la sortie de données sur l'imprimante ou au terminal.

AFFICHE SUR imprimante , FORMAT = MOD -1
terminal , MOD -2
MOD -3

ETAT = en-tête d'état

TPAGE = titre-de-page ...

TLIST = détail - *,détail - 2,détail - 3,détail - 4.

SUR imprimante indique que l'utilisateur aura le choix
terminal

entre une sortie à l'imprimante et une sortie à la machine à écrire. Ces deux noms seront remplacés par les codes mnémoniques qui désignent les deux supports dans le système d'exploitation employé.

L'utilisateur spécifiera le mode de format nécessaire par la clause FORMAT.

- en-tête d'état sera le titre de l'état imprimé.
Pour une commande AFFICHE, il n'y aura qu'un seul en-tête d'état. Cette clause ne sera pas utilisée en format de MOD - 3.
- titre de page: pour un même état, l'utilisateur peut prévoir trois titres de page au maximum. Le changement de titre se fera automatiquement au moment de l'impression. Dès que l'utilisateur a décrit les titres de page, le changement de valeur d'une rubrique de référence provoquera l'impression par le système du titre de page suivant. Les en-têtes de

colonnes que l'utilisateur désire voir imprimer seront produits automatiquement par le système.

- TLISTE: est la déclaration des données qui apparaîtront sur une ligne imprimée.
- détail - 1, détail - 2,détail - 4 sont les données qui apparaîtront sur une ligne imprimée. Elles seront soit extraites de fichiers, soit résultat de certains mouvements (opérations).

Les totaux intermédiaires et généraux seront produits automatiquement après chaque niveau de rupture (page, état), à condition que l'utilisateur ait manipulé au préalable la commande TOTAL.

6.3. Justifications et analyse des commandes choisies

Les différentes commandes du langage proposé ont été choisies en fonction des problèmes analysés qui se posent quotidiennement aux gestionnaires. Nous les avons regroupés en deux grandes catégories: les commandes relatives à la sélection des données et à leur extraction et les commandes relatives au traitement.

6.3.1. Commandes de sélection

Le but de la sélection est de repérer par leur identifiant, les informations qui répondent à des conditions portant sur une ou plusieurs caractéristiques. Ces commandes font appel à des routines (modules) de sélection qui permettent de réduire l'ensemble des données. Pour chacun des objets décrits, ces routines donnent la liste des données désignées par la question de l'utilisateur et effectuent sur cette liste des opérations logiques nécessaires pour obtenir finalement les seules données répondant à tous les critères de sélection.

- La commande DEMANDE

Cette commande permet de rechercher des informations désirées par l'utilisateur. Cette recherche se fait parmi les informations appartenant à un ou des fichiers ayant une structure imposée par la nature des applications. (a).examinées précédemment. Une exigence fondamentale due à la structure des fichiers étudiés et à la nature des problèmes posés est la suivante: la sélection des informations ne pourra se faire que sur toute une partie de la clé d'accès d'un objet. Chaque objet a, comme nous l'avons vu dans la partie structure, une ou plusieurs caractéristiques. Certaines de ces caractéristiques jouent le rôle de clé d'accès. Ainsi l'article budgétaire a pour clés: exercice, nature de budget, code économique, code fonctionnel, numéro d'ordre.

Dans le système que nous définissons, la sélection d'un ou plusieurs articles budgétaires se fera au moyen de critères arithmétiques pouvant être reliés par l'opérateur booléen ET.

Le premier critère arithmétique devra porter obligatoirement sur tout ou partie de la clé utilisée, ensuite seulement, on pourra utiliser des critères portant sur les autres caractéristiques d'un objet.

Exemple: L'utilisateur désire connaître pour un exercice donné, les crédits mis à jour alloués à une fonction. Cette question pourra se formuler par la commande "DEMANDE" comme suit:

DEMANDE TOUT CREDIT MIS A JOUR SI exercice=9 (4) ET fonction = 9 (3) (b). Ainsi formulée, cette commande est correcte.

Par contre, il serait incorrect d'écrire:

DEMANDE TOUT CREDIT MIS A JOUR SI fonction = 999 ET exercice = 9999; le critère "fonction" mis en premier lieu n'étant pas

(a) voir sous-schéma budget.

(b) Les chiffres remplaceront ces symboles dans un exemple réel; exercice signifiant l'année pour laquelle on désire connaître les crédits; il faudra fournir également le code fonction.

un élément de la clé d'accès.

Si pour le même exemple, l'utilisateur désire connaître les crédits mis à jour pour le budget ordinaire uniquement, la commande devient:

DEMANDE TOUT CREDIT MIS A JOUR SI EXERCICE = 9999 ET nature de budget = ordinaire ET fonction = 999; exercice et nature de budget étant des critères désignant la clé d'accès, ils doivent être écrits en premier lieu; il serait dès lors incorrect d'écrire:

DEMANDE TOUT CREDIT MIS A JOUR SI exercice = 9999 ET fonction = 999 ET nature de budget = ordinaire;

ou:

DEMANDE TOUT CREDIT MIS A JOUR SI fonction = 999 ET nature de budget = ordinaire et exercice = 9999.

- Justification du point de vue de l'utilisateur

Notre but est de fournir à l'utilisateur un langage d'utilisation qui lui permette d'exprimer de façon simple les traitements qu'il désire effectuer. En proposant une commande "DEMANDE", nous ne croyons pas nous être écartés de cet objectif. Cette commande traduit la logique d'un gestionnaire face à un problème. Si, indépendamment de toute référence à l'ordinaire, on demande à un utilisateur (gestionnaire): "quels sont les crédits mis à jour, il ne saurait répondre facilement à cette question, qui reste très floue dans sa formulation. Il peut rétorquer par des questions du genre: "veuillez préciser votre question, quels crédits mis à jour désirez-vous connaître et pour quel budget, pour quelle fonction ou pour quels articles budgétaires. On pourrait multiplier les exemples de ce type.

Il apparaît donc de manière claire que, même dans un système de gestion non automatisé, lorsqu'on pose une question à un

gestionnaire, ce dernier souhaite qu'on précise le champ d'application auquel on s'intéresse. La commande "DEMANDE" que nous avons proposée ne fait que traduire une logique tout à fait naturelle du gestionnaire. Elle peut se résumer en deux parties:

- DEMANDE valeur: précise les données ou les informations que l'on désire connaître.

- SI CRITERE de choix: détermine dans quelles catégories de données ayant même nature, on peut trouver les informations souhaitées, car celles-ci sont munies d'une structure sémantique et d'une structure d'accès logique.

D'une manière générale, et dans la mesure où il existe un certain ordre établi, pour retrouver les informations à un moment voulu, il y a trois façons de l'indiquer: (15)

- indiquer où se trouve l'information
- donner un chemin qui y conduit
- donner une partie du contenu de l'information cherchée.

C'est là toute la justification de la commande "demande". Nous croyons enfin que la formulation de la commande elle-même pourrait poser des problèmes insurmontables d'apprentissage à l'utilisateur, car les opérateurs logiques choisis ne sont pas propres à l'informatique mais sont également courants dans le domaine de la gestion.

En conclusion, la commande "DEMANDE" du point de vue informatique permet la sélection et l'extraction d'articles répondant à des conditions données. Ces éléments devront, suivant le cas, être édités ou subir au préalable certaines transformations.

Cet aspect du problème nous a conduit à séparer les commandes de sélection de celles relatives aux traitements. Pour les mêmes données, la sélection ne se faisant qu'une fois mais

- les traitements pouvant être diversifiés: - modifications
- sélection suivie de totalisation
 - transferts suivis d'édition
 - édition sous traitement, etc...

6.3.2. Commandes de traitement

Les commandes de traitement offrent à l'utilisateur la possibilité d'effectuer certaines opérations sur les éléments sélectionnés. Pour l'utilisateur, ceci est d'un intérêt évident; l'obtention des cumuls de certaines données, les "soldes" de certains éléments sélectionnés, certaines mises à jour urgentes, nécessitent l'existence d'opérateurs adéquats dans le langage. Nous avons proposé quelques opérateurs; cette liste n'a, bien entendu, aucune prétention exhaustive; elle tient uniquement compte des applications que nous avons décrites. Si les utilisateurs le souhaitent et en exprimant les besoins, il est possible de définir d'autres opérateurs supplémentaires.

- La commande TOTAL: Nous avons déjà explicité cette commande au paragraphe . Cette commande cite les données pour lesquelles on désire imprimer des totaux. L'échantillon de questions examiné précédemment, montre que cette commande est très utilisée pour la gestion des budgets provinciaux:

- totaux des crédits par Fonction
- totaux des crédits par code économique
- totaux des crédits budgétaires par service ou par section, etc...

Cette commande décharge le gestionnaire des problèmes de rupture et d'additions successives. En reprenant l'exemple de la page 70 , nous comprendrons mieux cela.

TOTAL section, service, PAR crédit analytique mis à jour:

Par cette commande, l'utilisateur désire obtenir les totaux de crédits analytiques mis à jour par section, par service et par budget analytique. Mais comme nous l'avons déjà dit, cette commande doit être précédée par une commande de sélection du genre:

DEMANDE crédit analytique mis à jour SI code service 999 ET
code service 999 ET exercice 1974.

Cette commande prépare les données qui seront disponibles pour les opérateurs de totalisations. Ces dernières ne pourront être effectuées que sur les données dont le code service est déterminé par les critères de sélection.

Dans cet exemple, les sous-totaux et totaux suivants seront faits

- sous total des crédits analytiques mis à jour par section. Il y aura une rupture à chaque changement de section;
- les sous-totaux des sections appartenant à un même service seront ensuite sommés pour obtenir un total par service. D'où une rupture après chaque service;
- les totaux des services considérés seront sommés pour obtenir un total général du budget analytique; ce dernier niveau n'aura pas de rupture car il n'existera qu'un seul total général.

Pour effectuer une totalisation, l'utilisateur se borne à paramétrer la commande "TOTAL"; les routines du système se chargeront des calculs nécessaires et de l'édition.

- Les commandes des mises à jour

Plusieurs applications de mise à jour ont été expliquées dans la première partie de ce travail. Les commandes MODIF et TRANSF ont été abordées au paragraphe 6.2. Notre langage étant conver-

sationnel, les mises à jour ne sont pas toujours aisées dans un mode "on line".

Les problèmes posés par un traitement "on line" des mises à jour sont généralement fort délicats, au fur et à mesure que le nombre de mises à jour effectuées par rapport au fichier initial augmente. L'organisation du fichier se détériore généralement et la consultation devient de moins en moins rapide... Le choix d'un processus de mise à jour "on line" sera toujours très onéreux et ne se justifiera que dans les cas où les contraintes de l'application l'imposent (15).

Les commandes de mise à jour seront utilisées pour les modifications ponctuelles. Les modifications plus importantes se feront en "batch".

- La commande "MODIF": permet de modifier une caractéristique d'un article budgétaire désigné par une commande de sélection. Les paramètres de la commande indiquent les caractéristiques que l'on souhaite modifier. Toutes les modifications budgétaires légales devront être opérées par l'intermédiaire de cette commande.

Du point de vue physique, cette commande doit assurer la recopie de l'article modifié à l'emplacement adéquat dans le fichier (a).

- La commande "TRANSF": est utilisée pour l'enregistrement des transferts budgétaires. Un transfert s'effectuant nécessairement entre un article "origine" et un article "destination", une commande de sélection (voire deux, si le transfert a lieu entre deux articles appartenant aux fichiers différents) désignera au préalable les articles concernés par le mouvement de transfert.

(a) Ceci est un aspect qui relève de l'implémentation.

Les modifications budgétaires et les transferts effectués seront suivis d'une édition en Mode 3 réalisée à l'aide de la commande "AFFICHE".

- La commande "SOLDE":

-Justification et syntaxe: pour pouvoir suivre l'évolution de la consommation des crédits qui leur ont été alloués, les chefs des services ont besoin de connaître les écarts entre les prévisions et les réalisations de leur budget. Pendant l'exercice budgétaire, en effet, il est intéressant de connaître la différence entre le crédit initial et le crédit mis à jour pour un poste donné du budget.

La commande "SOLDE" comme son nom l'indique, sert au calcul des soldes qui peuvent être calculés par le système et portés (édités) à la connaissance du responsable concerné.

La syntaxe de la commande est ainsi formulée:

SOLDE nom de donnée 1 , nom de donnée 2 .

Cette commande a pour effet de soustraire nom de donnée - 2 de nom de donnée - 1 et de placer le résultat dans une zone de mémoire appropriée en vue de l'édition.

Lors de l'utilisation de la commande AFFICHE, le contenu de cette zone sera automatiquement édité.

Pour pouvoir utiliser la commande "SOLDE" il faut au préalable sélectionner les données dont on souhaite connaître le solde par une commande "DEMANDE".

Exemple d'utilisation:

Question: pour pouvoir éditer un récapitulatif du budget, il faut calculer le solde entre dépenses et recettes de l'article "EXERCICES ANTERIEURS".

On écrira:

DEMANDE crédit mis à jour SI indicatif = 9(9) ET nature du

budget = recettes.

DEMANDE crédit mis à jour SI indicatif = 9(9) ET nature du budget = dépenses.

SOLDE crédit mis à jour, crédit mis à jour.

Le premier crédit mis à jour est celui sélectionné par la première commande DEMANDE. Le second est celui de la seconde commande.

- La commande AFFICHE:

La syntaxe de la commande "AFFICHE" a été présentée au paragraphe 6.2.5. L'utilisateur a le choix pour la sortie de ses résultats, entre plusieurs supports:

- terminal
- imprimante
- bande ou disque (édition différée).

La nature de différentes applications nous a amené à mettre à la disposition de l'utilisateur 4 types de format d'édition que nous résumons ainsi:

le format standard: utilisé pour l'édition de la totalité de l'information disponible sur le budget provincial ou analytique:

- édition du budget analytique pour un exercice déterminé
- édition du budget provincial sur un exercice
- édition des récapitulatifs.

Pour ce format, la sortie des résultats se fera soit sur imprimante, soit sur bande ou disque. Il existera une liste des modules d'éditions pour les différents états à sortir. Quand l'utilisateur voudra un certain type d'imprimé, il suffira de taper à la console une commande d'appel des modules appropriés en indiquant la valeur de différents paramètres. Cette façon

de procéder n'est pas une particularité de notre langage et les commandes d'appels des modules sont fournis par le système d'exploitation utilisé mais les modules auront été programmés à la demande de l'utilisateur.

6.33 Pourquoi des formats libres

Les formats libres ont pour but principal de produire à la demande, des informations adaptées à chaque niveau de responsabilité.

Ainsi, on doit mettre à la disposition d'un responsable du centre de frais, dans le plus grand détail, les éléments qui dépendent de sa gestion (budget qui lui est alloué en début d'année et rythme de consommation de celui-ci).

L'édition des états imprimés pose beaucoup de problèmes:

- édition d'en-têtes

Un en-tête d'état est l'en-tête de l'imprimé qui sera édité une seule fois pour un type d'imprimé.

Un en-tête de page sera édité systématiquement au début de chaque page de l'état ou à la suite d'un changement de valeur d'une rubrique de référence.

Un en-tête de colonne représente les titres des rubriques (a) à éditer

Description du contenu des lignes à éditer.

Description des variables physiques: ce sont des variables indiquant:

- le positionnement des lignes dans la page
- le maximum de lignes par page
- la gestion d'un compteur de ligne

(a) Plus petit élément d'informations sur lequel le système travaille.

- la gestion d'un compteur de pages
- la gestion de saut de page.

- Description des totalisations à réaliser

Les totaux devront être édités quand leur condition d'impression sera réalisée.

Toutes les variables nécessaires à une édition ne sont évidemment pas énumérées ici. Pour que le système que nous proposons puisse être utilisé par un non-informaticien, nous avons prévu dans la commande "AFFICHE" un minimum de variables paramétrables, les autres étant gérées automatiquement par le système.

- Choix d'un format par l'utilisateur

Les données à éditer seront sélectionnées au moyen de la commande "DEMANDE"; il sera possible également d'introduire dans les états imprimés, des éléments calculés au moyen des opérateurs simples (totalisateurs).

Pour la sortie des données en formats libres, 3 modes de formats ont été proposés. Ces 3 modes traduisent chacun un type de problème à éditer.

Le Mode - 1: destiné à fournir une réponse aux questions ayant un caractère global et général. Ce sera le cas pour une obtention d'informations relatives à tout le budget provincial ou tout le budget analytique d'un service, englobant toutes les natures de budget et natures d'articles.

Le Mode - 2: fournit des états en réponse aux besoins d'informations occasionnels ne nécessitant pas une recherche sur toutes les natures de charges ou toutes les natures de budget.

Le Mode - 3: fournit des réponses aux questions ponctuelles. L'imprimé ne comprend que quelques lignes au terminal. Ce mode

sera notamment utilisé pour vérifier les résultats des mises à jour effectuées "on line".

A présent, examinons plus en détail ces trois modes.

Le Mode - 1

La description de ce mode par l'utilisateur comprend les clauses suivantes:

- un en-tête d'état qui sera un nom que l'utilisateur veut attribuer à l'état à imprimer.

Exemple: s'il désire le titre "budget ordinaire", il écrira: EN-TÊTE d'ETAT= budget ordinaire. Pour simplifier le système, il n'y aura qu'un seul en-tête d'état pour une commande "AFFICHE". En d'autres termes, il sera incorrect d'écrire:

EN-TÊTE D'ETAT= budget ordinaire, budget extraordinaire, budget pour ordre.

- un en-tête de page qui sera un sous-titre d'état.

Exemple:dépenses. On écrira EN-TÊTE DE PAGE = dépenses.

On peut paramétrer 2 ou 3 en-têtes de page. Il sera dès lors correct d'écrire:

EN-TÊTE DE PAGE = dépenses, recettes.

Cette clause signifiera pour le système que les pages relatives aux dépenses seront éditées en premier lieu, ensuite, il y aura une rupture de page pour pouvoir commencer l'impression des pages relatives aux recettes. Ceci constitue une demande de saut de page implicite de la part de l'utilisateur.

- les en-têtes de colonnes: nous avons fixé le maximum de titres de colonnes à 4, après l'examen de questions que nous avons analysées. Mais, suivant les nécessités de gestion et la nature du support de sortie, le nombre peut

être augmenté ou diminué.

Exemple d'en-tête de colonne:

TCOLON= fonction, code économique, article, crédit initial.

Les en-têtes de colonnes ne sont pas décrits pour l'utilisateur. Le système les reproduit à chaque page de l'état sur base des paramètres de la clause `LIST`.

- les lignes à imprimer: les en-têtes de colonnes constituent les titres des données à éditer. En dessous de chaque en-tête de colonne, ligne après ligne, seront éditées les données extraites de fichiers ou résultant de certains calculs élémentaires.

- l'impression des totaux: celle-ci se fera à chaque changement de rubrique des références utilisées comme paramètres de la commande `TOTAL`. L'impression de tous les totaux se fera automatiquement, il suffira que l'utilisateur ait au préalable exécuté une commande `TOTAL`.

- le nombre maximum de lignes par page de l'état et la position de chaque ligne dans la page seront fixés par le système. L'utilisateur n'a aucune influence sur cela.

Le Mode - 2

Ce mode comprend à peu près les mêmes clauses que le Mode-1, la différence étant que ce mode n'accepte pas plus d'un type au niveau d'un état ou d'une page. Nous avons vu que le nombre d'en-têtes de pages en Mode - 1 pouvait être supérieur à 1. Il est évident que le Mode - 2 sera plus facile à programmer (moins de niveaux de rupture à prévoir). Il existe également d'autres différences dues à la nature des problèmes à résoudre par ces deux modes. Ce point a déjà été abordé.

Un seul en-tête d'état décrit par l'utilisateur. Cette clause est facultative pour le Mode - 2.

Il n'y aura qu'un seul en-tête de page par état. Il sera obligatoire et apparaîtra à toutes les pages imprimées.

Les en-têtes de colonne seront éditées à chaque page de l'état.

Quant aux lignes à imprimer, les données peuvent se limiter à une seule ligne.

Les totalisations intermédiaires et générales seront effectuées automatiquement, à condition qu'une commande TOTAL ait été utilisée auparavant.

Le nombre de lignes par page et les positions de ligne dans la page (interlignes) sont gérés par le système.

Le Mode - 3

Ce Mode est utilisé pour permettre des réponses aux questions ponctuelles, notamment lors des consultations après les mises à jour (modifications et transferts budgétaires légaux). Vu la quantité restreinte d'informations à sortir par commande, la commande AFFICHE pour ce format se réduira pratiquement à la description des lignes à imprimer:

- il n'y aura pas d'impression des totaux,
- les en-têtes d'état ne pourront être produits,
- les en-têtes de page n'existeront pas.

6.3.4. Messages du système

Pour un bon fonctionnement des opérations demandées par l'utilisateur non-informaticien utilisant le langage proposé, il est nécessaire que ce dernier soit guidé par l'ordinateur. Le langage étant conversationnel, le système enverra un message suivant les circonstances qui se présenteront.

Dans l'état actuel de l'étude, le nombre de messages possibles se limitera à trois, afin d'éviter les complications que cela poserait du point de vue de la réalisation pratique.

Ces messages, par convention, seront précédés d'un C pour les distinguer de commandes de l'utilisateur.

Ⓔ ACCEPTE: Ce message sera envoyé à l'utilisateur après toute commande bien exécutée par l'ordinateur. Chaque commande subira un contrôle de vraisemblance et un contrôle syntaxique. La réception de ce message au terminal signifiera pour l'utilisateur que l'opération qu'il a dernièrement demandée s'est bien déroulée et qu'il peut éventuellement passer une autre commande.

Ⓔ INVALIDE: Message du système à l'utilisateur lorsqu'une anomalie est dictée à la suite d'une commande. Le système par ce biais, demande une réaction de l'utilisateur qui pourra en l'occurrence resoumettre la commande à l'ordinateur après la correction appropriée.

Plusieurs cas classiques d'anomalies ou d'erreurs sont possibles.

Sans entrer dans les détails, deux types d'événements particuliers intéressent notre travail et devront être communiqués à l'utilisateur;

- les anomalies dues à une erreur de frappe au terminal;
- les anomalies dues aux erreurs syntaxiques dans l'utilisation du langage ou à l'inexistence de données demandées.

Dès que ces deux types d'erreurs se reproduiront, l'utilisateur recevra le message Ⓔ INVALIDE . Quant aux autres types d'anomalies ou d'erreurs dont les causes sont multiples, nous croyons qu'ils seront réglés par les informaticiens responsables du système d'exploitation, il serait exagéré de demander à un non-informaticien de corriger tous les types d'erreurs qui se présentent.

Ⓔ INDISPONIBLE: Ce message est une communication du système à l'utilisateur lorsque les données soumises au traitement demandé ne sont pas disponibles. Il est possible en effet, qu'un fichier ne soit pas disponible au moment où un utilisateur en a besoin. Un article de fichier peut être bloqué par un autre utilisateur qui effectue des mises à jour.

Ce message peut intervenir après une commande DEMANDE. A la réception de ce type de message, l'utilisateur devra attendre un peu avant de lancer une nouvelle commande DEMANDE.

6.3.5. Grammaire formelle du langage (a)

En utilisant les notations de Backus, les spécifications du langage peuvent s'exprimer ainsi:

```

<commande> : : = DEBUT <commande - 1> TERMINE
<commande - 1> : : = <commande - 2> / <commande - 1>
<commande - 2> : : = <fonction>
<fonction> : : = <demande> / <transf> / <modif> / <solde> /
                  <total> / <affiche>
<demande> : : = DEMANDE <caractéristique> SI <critère de choix>
<critère de choix> : : = <critère arithmétique> [ET <critère de
choix>]
<critère arithmétique> : : = <expression arithmétique> <opéra-
teur logique> <expression arithmétique-1>
<expression arithmétique-1> : : = { <caractéristique> }
                                   { <constante> }
<opérateur logique> : : = </> / <≤> / <=> / <≠> / <≥> / </>
<expression arithmétique> : : = <caractéristique>
<transf> : : = TRANSF caractéristique, caractéristique AVEC
numéro, montant, date
<modif> : : = MODIF caractéristique AVEC numéro, montant,
libellé, date
<solde> : : = SOLDE caractéristique, caractéristique
<total> : : = TOTAL caractéristique, caractéristique, caracté-
ristique PAR caractéristique

```

(a): les traits obliques dans la syntaxe doivent être inter-
prétés comme verticaux.

`<affiche> : : = AFFICHE SUR <support> , <format> <en-têtes>`
`<support> : : = <terminal>/<imprimante>/<mémoire périphé-`
`rique>`
`format : : = MODE - 1/ MODE - 2 /MOD - 3/MOD - S`
`<en-tête> : : = en-tête d'état , titre de page , <tlist>`
`<tlist> : : = caractéristique , <tlist> <caractéristique>`

Quelques remarques concernant la grammaire définie

- les identificateurs des caractéristiques signifient les noms des caractéristiques des fichiers, qui ont été définies lors de l'analyse de la structure sémantique.
 - Dans les commandes TRANSF et MODIF, numéro, montant, libellé, date, sont des noms de données. Nous les avons mis expressément dans ces deux commandes, car ces noms de données seront les seuls à pouvoir être utilisés dans ces commandes.
 - il est entendu que les différents supports peuvent être remplacés par leurs codes mnémoniques dans la commande AFFICHE. Par codes mnémoniques, il faut entendre les codes mnémoniques en usage dans le système d'exploitation utilisé.
 - Pour produire les différents groupes d'édition, il est nécessaire de résoudre le problème de compatibilité des zones, étant donné les différents formats des données susceptibles d'être utilisées.
- Chaque type de ligne à imprimer doit avoir un même format. Si l'en-tête définie par l'utilisateur dépasse la longueur maximum prévue par le système, il y aura troncature. En d'autres termes, chaque type de ligne à éditer aura un format unique et valable pour toute valeur présentée par l'utilisateur.
- Les intitulés de colonnes seront constitués de noms des données disponibles dans le dictionnaire des caractéristiques.

- Une sous-routine doit assurer les sauts de page adéquats sur base des indicatifs dont le changement de valeur devra être pris en considération.

6.3.6. Exemples de formulation des commandes

- Rechercher le crédit initial, le crédit mis à jour, les transferts d'un article du budget analytique.

DEBUT

DEMANDE article SI indicatif = X

AFFICHE SUR terminal, FORMAT= MOD - 3

TLIST = crédit initial, crédit mis à jour,
transferts

TERMINE

- Rechercher pour un service donné, le total des crédits par nature de charges.

DEBUT

DEMANDE article SI nom-du-service = X

TOTAL nature de charge, service PAR crédit
mis à jour

AFFICHE SUR imprimante, FORMAT = MOD - 2

TPAGE (a) = service

TLIST = code nature charge, crédit initial
TERMINE

- Editer le crédit total par service, par section pour une nature de charge.

DEBUT

DEMANDE article SI nature de charge = X

TOTAL nature de charge, section, service
PAR crédit mis à jour

(a) abréviation pour titre de page

AFFICHE SUR imprimante FORMAT = MOD - 1
EN-TETE D'ETAT = budget analytique
TPAGE = crédit par service
TLIST = crédit, nature de charge, nom de la section,
nom du service
TERMINE

- Edition du total de crédits par code économique pour
une fonction.

DEBUT
DEMANDE article SI fonction = X
TOTAL code économique, fonction PAR crédit mis
à jour
AFFICHE SUR terminal, FORMAT = MOD - 2
TPAGE= libellé fonction
TLIST = crédit mis à jour, code économique
TERMINE

- Edition pour un article du budget du crédit initial,
crédit mis à jour, les modifications budgétaires, les trans-
ferts, le solde non ventilé.

DEBUT
DEMANDE article SI libellé = X
AFFICHE SUR terminal, FORMAT= MOD - 3
TLIST = crédit initial, crédit mis à jour, solde
non ventilé, transferts
TERMINE

- quels sont les totaux de crédits d'une nature de
charge donnée pour un service?

DEBUT
DEMANDE article SI exercice= X ET code-de-nature
de charge = 9 (3)

DEBUT

DEMANDE article SI exercice = X ET code-de-nature de charge = Y (a)

TOTAL nature de charge PAR crédit mis à jour

AFFICHE SUR terminal, FORMAT = MOD - 2

TLIST = nature-de-charge, crédit mis à jour

TERMINE

- Réalisation d'une modification budgétaire sur un article:

DEBUT

DEMANDE article SI indicatif = X

MODIF article AVEC numéro, libellé, date

AFFICHE SUR terminal, FORMAT= MOD - 3

TLIST = numéro, montant, libellé article, date

TERMINE

± Effectuer un transfert entre deux articles.

DEBUT

DEMANDE article - 1 SI indicatif = X

DEMANDE article - 2 SI indicatif = Y

TRANSF article - 1, article - 2 AVEC numero, montant, date

AFFICHE SUR terminal FORMAT = MOD - 3

TLIST = numéro, montant de transfert, date

TERMINE.

Il est certain qu'au stade de la réalisation du langage, il sera préférable d'utiliser des mots plus courts ou des codes pour désigner les différentes commandes utilisées. Cela réduira considérablement les risques d'erreur, lors de l'utilisation des commandes.

Si l'on adopte une codification, il suffirait de remplacer

(a) Y sera remplacé dans un exemple réel par une occurrence de la caractéristique sur laquelle porte la condition.

les codes par les mots de commandes actuels.

6.4/ Considérations relatives à l'implémentation

L'implémentation et sa mise au point ne font pas l'objet de ce travail.

Nous nous sommes donc limités à présenter les idées générales concernant le sujet.

6.4.1. Solutions possibles

Pour exploiter les commandes du langage proposé, on est contraint de procéder à un travail de traduction effectué par un programme de traduction. Ce programme est plus ou moins complexe et son exécution nécessite un ou plusieurs passages en machine.

Pour transposer les commandes en un programme exécutable par l'ordinateur choisi, il existe 4 types de programmes de traduction: (13):

- les assembleurs
- les compilateurs
- les générateurs
- les interpréteurs.

- Les assembleurs traitent un langage proche du langage machine, et ayant dans sa forme élémentaire une correspondance instruction par instruction avec ce dernier: les langages d'assemblage ou autocodes, propres à un ordinateur donné.

Ces langages gardent la structure globale du langage machine tout en permettant l'adressage symbolique. Cette solution ne peut s'appliquer au langage que nous proposons, vu que nous le souhaitons proche du français.

- Quant à la solution consistant à décrire un compilateur pour la traduction du langage proposé, elle serait hasardeuse et

inutilement coûteuse. La tâche majeure d'un compilateur est d'établir une suite d'opérations dans un ordre correspondant à la logique de la machine à partir d'expressions dont l'ordre correspond à la logique du langage utilisé. Cela constitue un travail d'analyse très important, long à réaliser et très coûteux, qui ne se justifierait pas pour un langage simple comme celui que nous proposons.

- Une autre solution serait d'écrire un interpréteur qui traite le langage proposé en l'interprétant commande par commande au moment de l'exécution.

Dans la pratique, il semble que cette solution ne soit pas facile à réaliser.

- La dernière solution est d'écrire des programmes que certains appellent dans la littérature informatique, des générateurs.

Vus par les utilisateurs, les générateurs travaillent selon les mêmes principes que les compilateurs: en partant d'indications fournies dans le langage que nous proposons, c'est à dire de commandes, le générateur sélectionne des modules pré-programmés et les adapte au problème à résoudre grâce aux paramètres des commandes.

Vu que le langage que nous proposons a un vocabulaire très réduit et que sa traduction ne demanderait que très peu de traitement lexical et syntaxique, nous sommes favorables à cette dernière solution, même si la programmation de différents modules dans un langage de base ou évolué, peut être un travail de longue haleine.

C'est néanmoins une solution relativement moins chère et à risque d'erreur peu élevé, étant donné que les séquences pré-programmées sont déjà écrites et testées. Les seuls tests à réaliser pour ce générateur consisteront en la vérification de l'agencement de différents modules.

6.4.2. Méthode de conception

La solution que nous préconisons est d'écrire des modules qui traduiront chaque commande de l'utilisateur. Ces différents modules étant conçus de manière à être tout à fait indépendants les uns des autres (modularité). Ceci a pour avantage de permettre d'ajouter d'autres commandes sans avoir à modifier les commandes déjà existantes. Il est de ce fait aussi possible de modifier certaines commandes déjà existantes pour les adapter aux nouvelles exigences des gestionnaires sans devoir revoir toutes les commandes du langage. On peut également changer un module existant par un autre, plus performant, sans avoir à changer les autres modules.

Pour répondre aux requêtes (commandes) de divers utilisateurs, le système que nous proposons devra permettre:

- l'analyse syntaxique de chaque commande
- la réalisation des accès aux données
- l'exécution des opérations demandées
- l'édition des renseignements demandés.

Analyse syntaxique de chaque commande

Le but de cette analyse sera de vérifier si la commande employée par l'utilisateur est correcte. Par commande, il faut considérer non seulement le nom de celle-ci mais également les paramètres directifs qui l'accompagnent. Ces paramètres en effet, précisent notamment les articles ou les caractéristiques concernés.

Pour faire cette analyse, le système dispose de deux dictionnaires:

- le dictionnaire des commandes : c'est un lexique qui

contiendra toutes les commandes définies dans la grammaire du langage. A la lecture de chaque commande, le système consultera ce dictionnaire pour savoir si la commande existe. Pour toute commande mal écrite ou inexistante, le système enverra le message **INVALIDE** à l'utilisateur.

- le dictionnaire des caractéristiques: a pour but de vérifier si les données utilisées comme paramètres dans la commande existent dans les fichiers du système.

Comme pour les noms des commandes, si les données utilisées sont incorrectes ou inexistantes, le système envoie **INVALIDE** à l'utilisateur.

6.4.3. Réalisation des accès aux données

Cette phase est réalisée pour un module de sélection de données qui traduira une commande "DEMANDE". Le but de ce module est de repérer par leur identifiant, les objets qui répondent à des conditions portant sur une ou plusieurs de leurs caractéristiques et de les placer en mémoire centrale.

Cette sélection s'effectue en comparant la valeur d'une caractéristique appartient à un objet de fichier et une valeur de référence mentionnée par l'utilisateur par la commande "DEMANDE".

Cette recherche des caractéristiques désirées peut être réalisée de deux manières:

- sélection au moyen des valeurs des identifiants: c'est la recherche d'un enregistrement avec clé complète. Si les valeurs des caractéristiques utilisées dans la commande sont identiques à la clé d'accès du fichier, cela signifie que l'on peut accéder directement à l'occurrence de l'objet désiré.

Nous avons vu lors de l'étude de la structure d'accès, que l'identifiant pour un article budgétaire était composé de:

- code fonction

- code économique
- code de nature de charge
- exercice.

Tout accès direct aux occurrences recherchées se fera au moyen de ces identifiants. Mais la nature des applications, notamment lorsqu'on désire accéder à plusieurs occurrences d'un type d'objet, ayant une caractéristique commune, fera en sorte que l'on ne soit plus obligé de fournir une clé complète pour l'obtention des informations voulues, d'où le deuxième type de sélection:

- sélection avec clé partielle. Par exemple, l'utilisateur peut s'intéresser à tous les articles du budget appartenant à une même fonction, ou à une même nature de budget.

A la fin de l'exécution du module de sélection, on dispose en mémoire centrale d'occurrences d'objets à traiter.

6.4.4. Exécution des opérations demandées

A chaque opérateur prévu par la grammaire du langage, devra être associé un module. Les traitements demandés par l'utilisateur seront exécutés grâce à l'appel de ces modules. Ces derniers seront actualisés au moyen des paramètres figurant dans les commandes. Chaque module ainsi exécuté, fournira une ou plusieurs valeurs qui seront stockées dans les zones de mémoires connues du système.

- TOTAL: les résultats obtenus par le module de totalisation seront stockés pour préparer leur édition. Il y aura autant de zones de stockage qu'il y a de niveaux de ruptures prévus par la commande.

- MODIF: Essayons de dégager, sans entrer dans les détails, le processus qui sera suivi par le module qui traduira cette commande:

- retrouver la zone de mémoire contenant l'enregistrement à modifier. Cet enregistrement existe en mémoire depuis la phase de sélection;

- effectuer la modification nécessaire: remplacement du libellé ou le montant fourni comme paramètre dans la commande "MODIF";

- écriture de l'enregistrement sur disque ou bande à un endroit convenable;

- conservation en mémoire de l'enregistrement pour l'édition, étant donné que toute action de modification est suivie d'une édition.

Il est bien entendu possible de prévoir dans ce module un blocage de l'enregistrement concerné par la modification contre une mise à jour simultanée car cela entraînerait la création de résultats incohérents.

- TRANSF: Pour cet opérateur, il faut disposer d'au moins deux occurrences d'objets: un article-origine du transfert et un ou plusieurs-articles-destination du transfert.

Le module spécialisé dans l'exécution de cette commande doit pouvoir identifier en mémoire les zones contenant les articles nécessaires au transfert.

Les articles ainsi modifiés sont ensuite réécrits à l'emplacement convenable dans le fichier. Toute action de transfert sera suivie d'une édition.

- SOLDE: Le module de calcul des soldes aura pour tâches principales:

- l'acquisition des valeurs des opérantes;

- la réalisation de la soustraction et le stockage du résultat dans une zone de mémoire d'où il sera extrait pour l'édition.

6.4.5. Définition des états imprimés

Les états imprimés budgétaires sont fournis par le système sous forme de tableaux. Un tableau comprend un certain nombre de colonnes plus le titre de l'état, le titre de page et l'intitulé des colonnes.

Etant donné que la structure de l'état est définie par l'utilisateur au moyen de renseignements figurant dans la commande "AFFICHE", le système, sur base de ces renseignements, fournit en sortie un texte source dans un langage évolué, contenant la description de l'état ainsi que les traitements à accomplir pour la production de la sortie.

Il y aura un module d'édition pour la production de chaque état standard. Quant aux formats libres, un module d'édition est prévu pour chaque mode de format, il y en aura dès lors trois.

Constitution des lignes à éditer

L'édition de l'imprimé se fera par groupe d'édition. Par groupe d'édition, il faut entendre une liste ordonnée de types de lignes qui admettent la même condition d'impression. D'une façon générale, on distinguera les groupes d'édition:

- en-tête d'état
- en-tête
- détail
- total
- total général.

CONCLUSIONS

L'objectif de ce mémoire a été de décrire un langage permettant aux gestionnaires provinciaux de réunir rapidement toutes les informations dont ils peuvent avoir besoin, concernant leur budget.

Pour compléter notre étude, il était nécessaire d'aborder les aspects suivants:

- la description de l'implémentation du langage;
- la mise au point de cette implémentation;
- la réalisation des tests du langage permettant de vérifier si les objectifs attendus ont été atteints, à savoir l'assimilation aisée du langage par un non-informaticien, l'extensibilité de ce langage et son adaptation à l'aide à la décision.

Dans l'état actuel de notre travail, les spécifications du langage sont des propositions qui méritent d'être approfondies.

Toutefois, en tant qu'instrument d'aide à la décision, le langage que nous proposons est conçu pour traiter des questions qui ne sont pas prédéfinies, mais qui restent dans le champ d'application relatif aux budgets de la Province.

Dans le cas où ce champ s'avère restreint, la modularité des commandes permet d'en augmenter le nombre sans modifier les commandes existantes.

Enfin sur le plan des possibilités offertes par le langage, il est certain que seule une utilisation effective nous aurait permis de juger de l'intérêt des options que nous avons choisies et de la nécessité de les conserver ou de les modifier.

Références

- (1): Arrêté royal portant règlement général sur le contrôle des engagements de dépenses et sur la comptabilité des Provinces.
- (2): Finances Publiques par Fernand Baudhuin - Bruxelles
Etablissements Emile Buyland
- (3): Comptabilité analytique d'exploitation par A. Rapin et J. Poly - DUNOD 1972
- (4): Gestion budgétaire par J. Meyer - DUNOD
- (5): Système Informatique de la Province de Namur.
1. Engagements. Description par le Centre d'Informatique de Gestion et de Recherche (CIGER)
- (6): Banques de données. 1ère et 2ème partie par A. Cabanès,
Conservatoire National des Arts et Métiers
- (7): Système de conception et d'exploitation d'une base de données - Institut d'Informatique, Facultés Universitaires Notre-Dame de la Paix, Namur
1ère partie
- (8): Colloques IRIA - Afcet Iria - banque de données AIX-EN-PROVENCE 4.5.6. juin 1973
- (9): CII carnet de notes - SOCRATE - SPECIFICATIONS DE DEFINITION
- (10): Interactive Query Facility (IQF) for IMS/360 version 2
General Information Manual (Program Product) IBM
- (11): Budgets provinciaux à classification fonctionnelle et économique . Références 533/B/DG (Circulaire Ministère de l'Intérieur, Direction générale de la Gestion financière des institutions régionales locales)

- (12): Compte-rendu des réunions en vue de la création d'un système de gestion intégrée des finances provinciales - Rapporteur H.CEIEGHIN août 1974 (FECRIW)
- (13): PRINCIPES de Programmation des ordinateurs , par Annette LAURET
- (14): CODACYL COBOL: DATA BASE FACILITY PROPOSAL (OTAWA 1972) GOV OF CANADA
- (15): Cycle de formation approfondie à l'Analyse Fonctionnelle Cours de: Organisation des fichiers - C.CHERTON, Professeur aux Facultés Universitaires de Namur.

Table des fonctions.

A03	B.O. Exercices antérieurs
E03	B.E. Exercices antérieurs
000	Général
010	Dette générale
020	Fonds
040	Impositions et redevances
050	Assurances
101	Autorités provinciales
104	Administration provinciale
105	Cérémonial officiel
106	Formation administrative
121	Appareil fiscal et financier
124	Patrimoine privé
136	Parc automobile
137	Service technique des bâtiments
320	Justice - Police
420	Communications routières
480	Hydraulique
511	Etude des zonings industriels et commerciaux
521	Commerce et artisanat
530	Industrie
550	Production et transport d'énergie
560	Tourisme
600	Agriculture - Horticulture
623	Agriculture
700	Enseignement général
732	Enseignement agricole
735	Enseignement technique
752	Enseignement spécial
761	Formation de la jeunesse

762 Culture et Loisirs
770 Beaux Arts - Archéologie - Protection de la nature
790 Cultes
801 Dépenses générales pour interv. sociales
833 Assistance Sociale
840 Aide sociale
843 Aide familiale
860 Relations du travail et protection du travail
870 Hygiène publique et santé publique
872 Etablissements de soins
877 Eaux usées
922 Logement

Code économique : validité suivant les valeurs de la table
ci-dessous:

Table des codes économiques.

060 Prestations
061 Transferts
062 Dette
063 Total
070 Personnel
071 Fonctionnement
072 Transferts
07X Dette
073 Total
080 Transferts
081 Investissements
082 Dette
083 Total
090 Transferts
091 Investissements
092 Dette
093 Total
990 Total exercices antérieurs

X

BUMP



0 0 7 8 5 6 7 0 0

*FM B16/1977/17

